



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

**POSOUZENÍ INFORMAČNÍHO SYSTÉMU A
NÁVRH ZMĚN**

INFORMATION SYSTEM ASSESSMENT AND PROPOSAL FOR ICT
MODIFICATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Hana Puklická

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Aleš Klusák, Ph.D.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav informatiky
Studentka: **Bc. Hana Puklická**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **Ing. Aleš Klusák, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Posouzení informačního systému a návrh změn

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je posouzení a analýza stavu informačního systému TimeMission ve společnosti ARTIN spol.s.r.o. Na základě provedené analýzy a celkového zhodnocení stavu systému, budou navržena možná řešení, která by měla vést ke zmírnění slabých stránek či nedostatků systému a tím zvýšit jeho efektivnost.

Základní literární prameny:

BASL, J a R. BLAŽÍČEK. Podnikové informační systémy. 2. vyd. Praha: Grada Publishing, 2007. 288 s. ISBN 978-80-247-2279-5.

BLAŽKOVÁ, M. Marketingové řízení a plánování pro malé a střední firmy. Praha: Grada Publishing, 2007. 278 s. ISBN 978-80-247-1535-3.

HORÁK, J. a M. KERŠLÁGER. Počítačové sítě pro začínající správce. 3. aktual. vyd. Brno: Computer Press, 2006. 212 s. ISBN 80-251-0892-9.

MOLNÁR, Z. Moderní metody řízení informačních systémů. 1. vyd. Praha : Grada Publishing, 1992. 347 s. ISBN 80-85623-07-2.

NEUWIRTH, B. Problematika hodnocení optimality a vyváženosti podnikových IS. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2009. 150 s.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd. Brno: ComputerPress, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 28. 2. 2017



doc. RNDr. Bedřich Půža, CSc.
ředitel



doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Diplomová práce se zabývá analýzou informačního systému Time Mission ve společnosti ARTIN, spol. s.r.o. Hlavní část práce je tvořena analýzou systému včetně popisu síťové infrastruktury a technologií, které přispívají k vývoji analyzovaného systému. Na základě provedených analýz budou navrženy možné změny, které by měly vést ke zvýšení efektivnosti informačního systému.

Abstract

This diploma thesis focuses on the information system Time Mission owned by ARTIN Ltd. The main part of the thesis is composed of analysis of the system including description of the network infrastructure and used technologies which are useful for another development of software. On the basis of analysis possible changes will be proposed. These changes should lead to increase in efficiency of the information system.

Klíčová slova

Time Mission, informační systém, IS, software, analýza, strategie, testování.

Keywords

Time Mission, information system, IS, software, analysis, strategy, testing.

Bibliografická citace

PUKLICKÁ, H. *Posouzení informačního systému a návrh změn*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 90 s. Vedoucí diplomové práce Ing. Aleš Klusák, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracovala jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušila autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 24.5.2017

Poděkování

Tímto bych ráda poděkovala mému vedoucímu diplomové práce panu Ing. Alešovi Klusákovi, Ph.D. za cenné rady a poznámky, při zpracování této práce. Dále bych ráda poděkovala paní Ing. Lucii Smrčkové za oponenturu této práce a panu Ing. Eduardovi Tomkovi za poskytnutí zásadních informací o analyzovaném systému.

Obsah

Úvod.....	12
1 Cíle a metodika práce	13
2 Teoretická východiska práce	14
2.1 Data	14
2.2 Informace	14
2.3 Znalosti.....	14
2.4 Systém	15
2.5 Informační systém.....	15
2.6 Funkce informačního systému	15
2.6.1 Transakční systémy.....	15
2.6.2 Informační systémy pro řízení	16
2.6.3 Systémy pro podporu rozhodování	16
2.6.4 Systémy pro podporu vrcholového řízení	16
2.7 Životní cyklus informačního systému.....	16
2.8 Uživatelé informačního systému.....	17
2.8.1 Enterprise Resource Planning.....	18
2.8.2 CRM-Customer Relationship Management.....	18
2.8.3 SCM – Supply Chain Management	19
2.8.4 Business Intelligence	19
2.9 Zajišťování a řízení kvality	20
2.10 Verifikace a validace	20
2.11 Úrovně a základní typy testů	23
2.11.1 Testování jednotek.....	23
2.11.2 Integrační testování.....	23
2.11.3 Systémové testování	24

2.11.4	Akceptační testování.....	24
2.12	Smoke a sanity testy	24
2.13	SWOT analýza.....	25
2.14	Analýza vnitřního prostředí	25
2.15	Metoda HOS2009	26
2.15.1	Oblasti hodnocení IS metodou HOS2009.....	26
2.16	Předaplikační fáze	27
2.17	Aplikační fáze.....	28
2.18	Fáze tvorby výstupů	28
2.19	Fáze zpracování výstupů	29
2.20	Fáze interpretace výsledků	30
3	Analýza současného stavu	32
3.1	Základní údaje o společnosti.....	32
3.2	O společnosti.....	32
3.2.1	Organizační struktura.....	32
3.2.2	Analýza interních faktorů	33
3.3	SWOT analýza informačního systému.....	35
3.4	Servery	37
3.5	Síť.....	38
3.6	Logické zapojení a topologie sítě.....	38
3.7	Zálohování aplikačních dat a serverů.....	39
3.8	Monitorování sítě, hardwaru a služeb	40
3.9	Analýzovaný systém Time Mission	40
3.9.1	Architektura a přístup do systému	40
3.9.2	Oprávnění.....	41

3.10	Serverové skriptovací technologie pro dynamické generování HTML stránek využívané v Time Mission.....	42
3.11	Moduly systému	43
3.12	Issue Tracking Systém.....	48
3.12.1	JIRA software v6.3.6	48
3.12.2	Životní cyklus ohlášeného ISSUE	48
3.13	Interní systém znalostí.....	49
3.13.1	Confluence	49
3.14	Version control systém (VCS).....	50
3.14.1	GitLab	51
3.15	Continuous Integration	52
3.15.1	JENKINS	53
3.16	Metoda HOS2009 hodnocení optimality a vyváženosti IS.	55
3.16.1	Předaplikační fáze.....	55
3.16.2	Aplikační fáze	56
3.17	Popis jednotlivých zkoumaných oblastí systému	59
3.17.1	Fáze tvorby výstupů.....	62
3.17.2	Fáze zpracování výstupů.....	63
3.18	Vyhodnocení metody HOS09.....	65
4	Vlastní návrhy řešení	68
4.1	Doporučené strategie na základě provedené metody HOS09	68
4.2	Návrhy změn na základě SWOT analýzy a vlastního pozorování	70
4.2.1	Code review	70
4.2.2	Automatické testování a integrace testů do JENKINS	72
4.2.3	Dokumentace testů za pomoci snímkování	76
4.2.4	Unit testy.....	76

4.2.5	Kontejnerová virtualizace	79
4.2.6	Multijazyčnost	82
4.3	Odhad nákladů	82
Závěr		84
Seznam použité literatury		85
Seznam obrázků		87
Seznam tabulek		88
Seznam vzorců		88
Seznam grafů		88
Seznam příloh		89

Úvod

V rozhodovacím procesu je pro řídící pracovníky důležité mít k dispozici včas a v dostatečné kvalitě informace, které jsou pro podnik klíčové. Je tedy podstatné, aby takovéto informace byly uchovávány a zpracovávány patřičným způsobem.

Informační systémy jsou tedy tou správnou cestou, jak přehledně evidovat informace napříč všemi interními procesy společnosti a posléze tak z nich získávat potřebné výstupy. Je nutné tyto systémy udržovat a vylepšovat tak, aby fungovaly, jak byly navrženy, a poskytovaly tak jeho uživatelům při jejich práci co největší komfort. Myšleno je tím především zvýšení efektivity jejich práce a konkurenceschopnosti podniku v rámci trhu. Pro úspěšný chod informačního systému je důležitý i hardware, na kterém systém běží včetně počítačové sítě.

Důležitá je nejen jeho údržba, ale také neustálé zlepšování a zdokonalování systému, především tehdy pokud firma rozšiřuje svoji činnost, nebo narůstá počet jejích klientů a tím i projektů. S tím souvisí rostoucí nároky na informační systém a jeho funkcionalitu, kterou je třeba rozšířit nebo zcela obměnit.

Jedním z trendů poslední doby je zabezpečený přístup do informačního systému přes webové rozhraní, tedy přes tenkého klienta, kde samotná aplikace není nainstalována na klientském počítači, ale běží přímo na serveru. Tato výhoda umožňuje pracovníkům a především managementu společnosti, mít neustálý přehled o chodu společnosti odkudkoliv, kde je umožněno připojení na internet.

Pokud se firma bude zaměřovat na slabé stránky systému a bude se je snažit vylepšit nebo zcela odstranit, bude pro firmu snadnější držet krok se svými konkurenty a rozšiřovat svůj potenciál.

1 Cíle a metodika práce

Cílem této práce je posouzení a analýza stavu informačního systému Time Mission ve společnosti ARTIN spol. s. r.o. Na základě provedené analýzy a celkového zhodnocení stavu systému, se pokusím navrhnout možná řešení, která by měla vést ke zmírnění slabých stránek či nedostatků systému a tím zvýšit jeho efektivnost.

K samotné analýze využiji metodu HOS09 a SWOT analýzu systému. Rovněž provedu i posouzení vlastním pozorováním, při kterém využiji své znalosti se systémem, se kterým ve společnosti běžně pracuji.

Na základě výše zmíněných analýz se pokusím najít slabá místa systému a především navrhnout změny, které povedou k eliminaci problému, zvýšení uživatelského pohodlí při práci se systémem a v neposlední řadě i případné zvýšení efektivnosti analyzovaného systému.

2 Teoretická východiska práce

2.1 Data

Data si lze představit jako prosté zobrazení skutečnosti. Sama o sobě nemají dobrou vypovídající hodnotu. Lze je však považovat za potenciální zdroj informací.

Data může člověk uložit pro pozdější zpracování, transformovat je do jiné podoby, například zaznamenat na papír nebo elektromagnetické zařízení. (1)

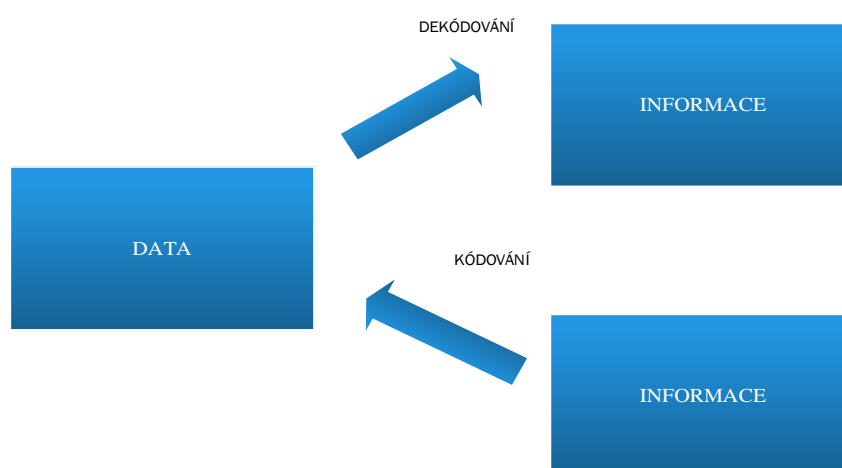
2.2 Informace

Informacemi se rozumí data, kterým uživatel přisuzuje určitý význam a který má pro příjemce objektivní informační hodnotu. Nositelem informace mohou být číselná data, obraz, zvuk případně další smyslové vjemy. Na rozdíl od dat nemůžeme informaci skladovat. Na druhé straně informace jako zdroj poznání, jsou zdrojem obnovitelným a nevyčerpatelným. Přestože, má informace nehmotný charakter, je spojena s určitým fyzickým pochodem, který ji nese. (1)

2.3 Znalosti

Robert M. Hayes: „Znalosti jsou výsledkem porozumění informací, která byla právě sdělena a její integrace s dřívějšími informacemi“. (2, s.13)

Znalosti můžeme chápat jako využití jiných informací a dat v různých situacích, tak aby byly přínosem v procesu rozhodování. (2)



Obrázek 1: Vztah dat a informací (Zdroj: (2) Vlastní zpracování)

2.4 Systém

V současné době se pojem systém užívá jako označení určité části reálného světa s charakteristickými vlastnostmi. Tyto systémy se dělí na systémy přirozené, které nejsou vytvořeny člověkem a systémy umělé, které jsou vytvořeny člověkem. Systém lze definovat mnoha způsoby, podle Charlese S. Wassona lze definovat systém následovně. (3)

„Systém je integrovaná sada prvků, z nichž každý má explicitně stanovené a ohraničené možnosti. Tyto prvky pracují synergicky, aby vytvořily hodnotný proces, který umožní uživateli uspokojit potřeby v daném operačním prostředí konkrétním výsledkem a pravděpodobnosti úspěchu.“ (3, s.18)

2.5 Informační systém

„Informační systém lze definovat jako soubor lidí, metod a technických prostředků zajišťující sběr, přenos, uchování a zpracování dat s cílem tvorby a poskytování informací dle potřeb příjemců informací činných v systémech řízení.“ (4,s.12)

2.6 Funkce informačního systému

Možnosti a funkce informačního systému závisí na potřebách daných firem. V dnešní době je převážná většina potřeb řešena za pomoci softwaru. Mnohé firmy, které pracují v určitém oboru či mají specifické požadavky, mohou využít možnosti vytvoření informačního systému na míru.

Informační systém lze klasifikovat různě. Funkce informačního systému a systému řízení jsou úzce propojeny. S vyšší úrovní řízení roste jistá neurčitost v požadavcích na IS, a zároveň se zmenšuje objem získaných informací v důsledku jejich selekce a agregace. Současně tak roste potřeba externích informací z podstatného okolí firmy. Informační systém firmy můžeme členit na určité části, z nichž každá z nich plní určitou funkci.

Pro každou takovou úroveň, pak existuje softwarová aplikace, která plní požadované funkce dané úrovně. (4)

2.6.1 Transakční systémy

Transakční systémy (Transaction Processing Systems-TPS) tyto systémy slouží pro operativní úroveň řízení. Ukázkou využití transakčních systémů může být například

zpracování transakcí objednávky zboží, prováděné pracovníkem obchodního oddělení.
(5)

2.6.2 Informační systémy pro řízení

Informační systémy pro řízení (Management Information Systems-MIS) vycházejí z ekonomických a účetních systémů a uživatelé v nich vyhledávají informace, které je zajímají. MIS se využívají především pro taktickou úroveň řízení. Z těchto systémů lze vytvářet potřebné reporty. Například periodické výstupy z transakčních systémů. (5)

2.6.3 Systémy pro podporu rozhodování

Systémy pro podporu rozhodování (Decision Support Systems-DSS) pomocí těchto systémů lze vytvářet rozmanité analýzy dat, bez potřeby složitého ovládání. Jedná se především o počítačovou podporu metod pro rozhodovací analýzy a operační systémové analýzy. Umožňují rychle provádět příslušné výpočty a manipulace se vstupními daty. Na druhé straně předpokládají, že uživatel je schopný porozumět podstatě metody a ví, kdy a proč jí má použít a jaká vstupní data musí zajistit, aby ji mohl aplikovat na svůj problém. (5)

2.6.4 Systémy pro podporu vrcholového řízení

Systémy pro podporu vrcholového řízení (Executive Information Systems-EIS) Slouží převážně vrcholovému vedení organizace, které se zajímá hlavně o informace z okolí společnosti. EIS poskytuje přístup jak k externím informacím, tak propojením s informačním systémem společnosti. Systémy umožňují multidimenzionalitu, která umožňuje jednoduše a rychle vytvářet pohledy na data, jejich řazení do souvislostí, indikací odchylek klíčových ukazatelů od plánovaných hodnot, práci s historickými daty a i budoucím vývojem. (5)

2.7 Životní cyklus informačního systému

Životní cyklus každého informačního systému je tvořen životními fázemi, které popisují jeho vývoj v čase od jeho vzniku a nasazení až po jeho ukončení užívání.

Životní cyklus se dle většiny autorů rozděluje do následujících fází. (6)

Analýza a návrh- definice požadavků a návrh systému až na úroveň, kdy je možné začít navržený systém implementovat.

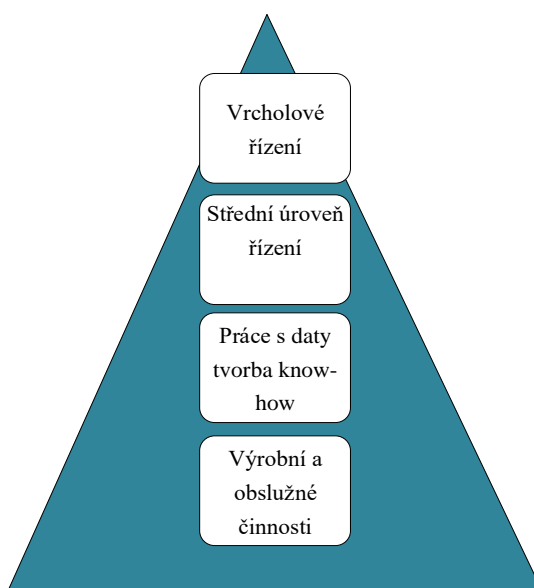
Implementace-vytvoření fungujícího systému, realizace v daném implementačním prostředí. Systém musí bezchybně fungovat a musí mít implementované všechny stanovené požadavky.

Zavedení do provozu-instalace technického a programového vybavení, školení uživatelů, vytvoření a úpravy databáze, integrační a zátěžové testy.

Provoz, údržba, podpora a „odchod do důchodu“- zajištění provozu systému, jeho údržba a rozvoj vzhledem k novým uživatelským požadavkům. Monitorování provozu, návrh změn a úprav. Jsou integrovány další aplikace, které detailněji pokrývají klíčové procesy, případně jsou nasazeny také proto, že původní informační systém již nedokáže pokrýt nové procesy v dané oblasti. (6)

2.8 Uživatelé informačního systému

Uživatele informačního systému lze rozdělit do několika úrovní napříč celou organizací. Jednotlivé úrovně jsou znázorněny na následujícím **obrázku č. 2.** (1)



Obrázek 2: Čtyřvrstvá organizační pyramida z pohledu práce IS/IT v podniku (Zdroj:(1) Vlastní zpracování)

Holisticko-procesní klasifikace člení podnikové IS podle jejich praktického využití. Podle této klasifikace je podnikový informační systém tvořen několika oblastmi. (7)

ERP - zaměření na řízení interních procesů

CRM – systém spravující procesy směřované ke spolupráci se zákazníky.

SCM – systém řídicí dodavatelský řetězec

BI – Business Intelligence

2.8.1 Enterprise Resource Planning

Informační systémy ERP představují prostředky pro řízení a plánování interních procesů podniku. Zajišťují automatizaci a integraci důležitých procesů společnosti. Mezi procesy řídicí ERP lze zařadit výrobu, ekonomiku, lidské zdroje, nákupní, prodejní a výrobní (vnitřní) logistiku.

Správně zvolený ERP systém přináší výraznou výhodu před konkurencí. Společnost velmi rychle získává podrobné informace o výkonnosti podniku i jejích obchodních partnerech. ERP systémy se rozdělují podle oborového a funkčního zaměření. (7)

All-in-one nabízí široké a detailní pokrytí všech klíčových podnikových procesů. Příkladem mohou být systémy typu Microsoft Dynamics NAV, SAP Business Suite a Oracle E-business Suite.

Best-of-breed zaměřují se na specifické procesy nebo obory.

Lite ERP odlehčená verze určené především pro malé a střední podniky. Systémy mají omezenou funkcionalitu a jsou cenově dostupnější. (7)

Hlavní požadavky ERP systému

- Realizace měřitelných přínosů v oblasti snižování nákladů vznikajících při neefektivním řízením podniku.
- Realizace neměřitelných přínosů v oblasti řízení podnikových procesů a dostupnosti informací v reálném čase. (7)

Hlavní vlastnosti ERP systému

- systém umožňuje automatizovat a integrovat hlavní podnikové procesy
- umožňuje sdílení dat, postupů a určuje standardy pro celý podnik.
- systém umožňuje vytvářet a zpřístupnit informace v reálném čase.
- umožňuje získávat historická data. (7)

2.8.2 CRM-Customer Relationship Management

Jedná se o systém zaměřený na řízení vztahů se zákazníky. Tento systém třídí a shromažďuje údaje o zákaznících, jejich kontakty, probíhající obchodní procesy a dosahované tržby. Lze podle nich vyhodnocovat veškeré obchodní činnosti v rámci společnosti. Cílem tohoto systému je identifikace potřeb zákazníků, zlepšení služeb, které

jim podnik nabízí a v neposlední řadě lépe porozumět zákazníkům. CRM systém je postaven na operativním řízení a rozboru zpracovaných dat. Mezi těmito prvky je nutná vzájemná propojenost a vyváženost. (1)

Hlavní procesy CRM

- **Řízení kontaktů** – představuje vícekanálovou komunikaci se zákazníky, a to jak v interním, tak i externím okolí firmy.
- **Řízení marketingu** – zabývá se řízením marketingových zdrojů, plánováním a realizací a vyhodnocováním marketingových kampaní.
- **Řízení obchodu** – zahrnuje objednávkový cyklus, řídí kontakty, zaznamenává vytváření nových objednávek a jejich převzetí zákazníkem.
- **Servisní služby** – mají za úkol zajistit záruční i pozáruční servis, jejich cílem je posilovat zákaznickou spokojenost. (7)

2.8.3 SCM – Supply Chain Management

Řídí dodavatelsko-odběratelské vztahy podniku. Jde o logistický proces, který zabezpečuje a řídí skladování produktů a řízení služeb s nimi spojenými, tak aby vedl k uspokojení zákazníků. Systémy SCM představují komunikaci dodavatelů a odběratelů pomocí informačních systémů, sdílení informací a plánování celkového postupu během zpracování zakázky. Prostřednictvím propojení a výměny informací mohou partneři v rámci řetězce spolupracovat, plánovat a koordinovat celkový postup a tím zvýšit úroveň celého procesu. (1)

Metody řízení uplatňované v rámci SCM

- Systém plynulého zásobování
- Řízení zásob dodavatelem
- Efektivní reakce na požadavky zákazníka
- Společné plánování a predikce v dodavatelském řetězci. (1)

2.8.4 Business Intelligence

Proces, při kterém probíhá transformace **dat** na **informace** a **informace** na **poznatky**. Vstupem jsou velké objemy dat a výstupem předem neznáme poznatky, které se dále využívají v procesu rozhodování. Data jsou uložena v multidimenzionální podobě v datových skladech a jsou získávána pomocí OLAP analýzy či data miningu.(1)

2.9 Zajišťování a řízení kvality

Z anglického Software Quality Assurance-SQA.

Zajišťování kvality softwaru je zaměřováno na kvalitu procesů celého životního cyklu softwaru. Mezi hlavní aktivity tak patří definice, zavedení procesů včetně používaných norem, procedur, metrik, nástrojů a následně kontrola jejich dodržování a hodnocení s cílem nalézt případné zlepšení. Zajišťování kvality se tak snaží primárně předcházet vzniku defektů.

Řízení kvality softwaru (Software Quality Control) je zaměřeno na výstupy z jednotlivých procesů, u kterých ověřuje, zda odpovídají specifikacím a požadavkům. Řízení kvality je zaměřeno na nalezení defektů, jejich odstranění a následné ověření správnosti, již aplikovaných změn. (8)

2.10 Verifikace a validace

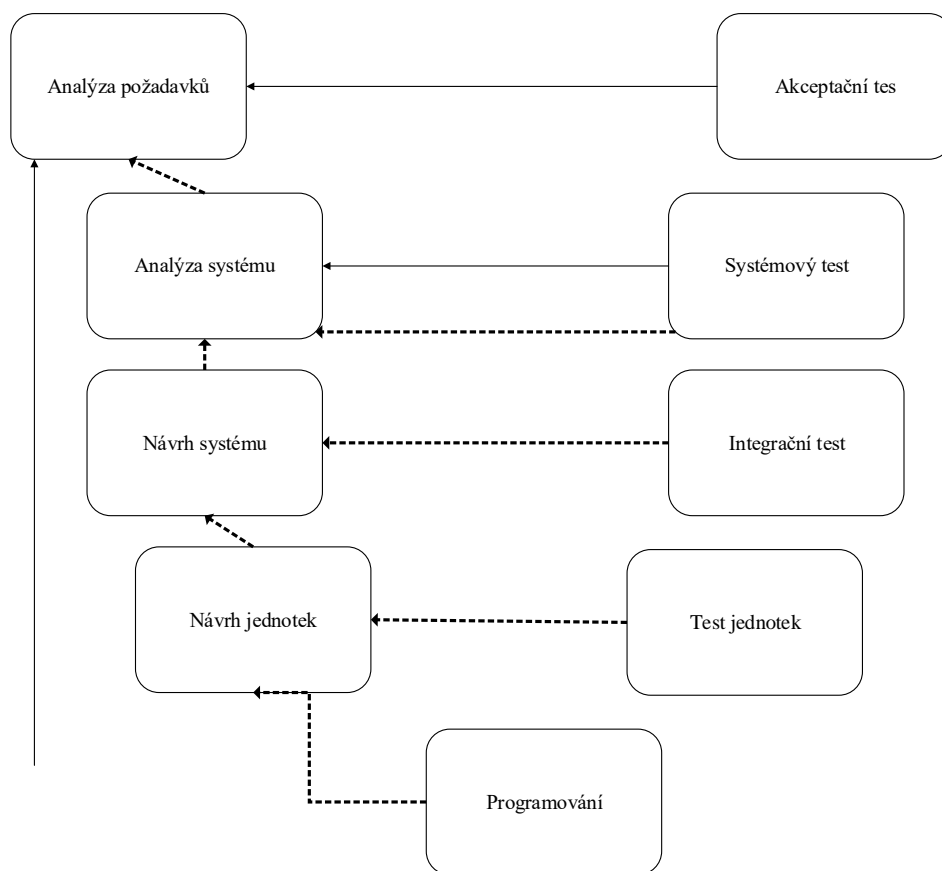
Verifikační a validační aktivity ověřují, zda software odpovídá specifikacím a jako celek poskytují to, co uživatel potřebuje.

- Verifikace: Vytváříme produkt správně?
- Validace: Vytváříme správný produkt?

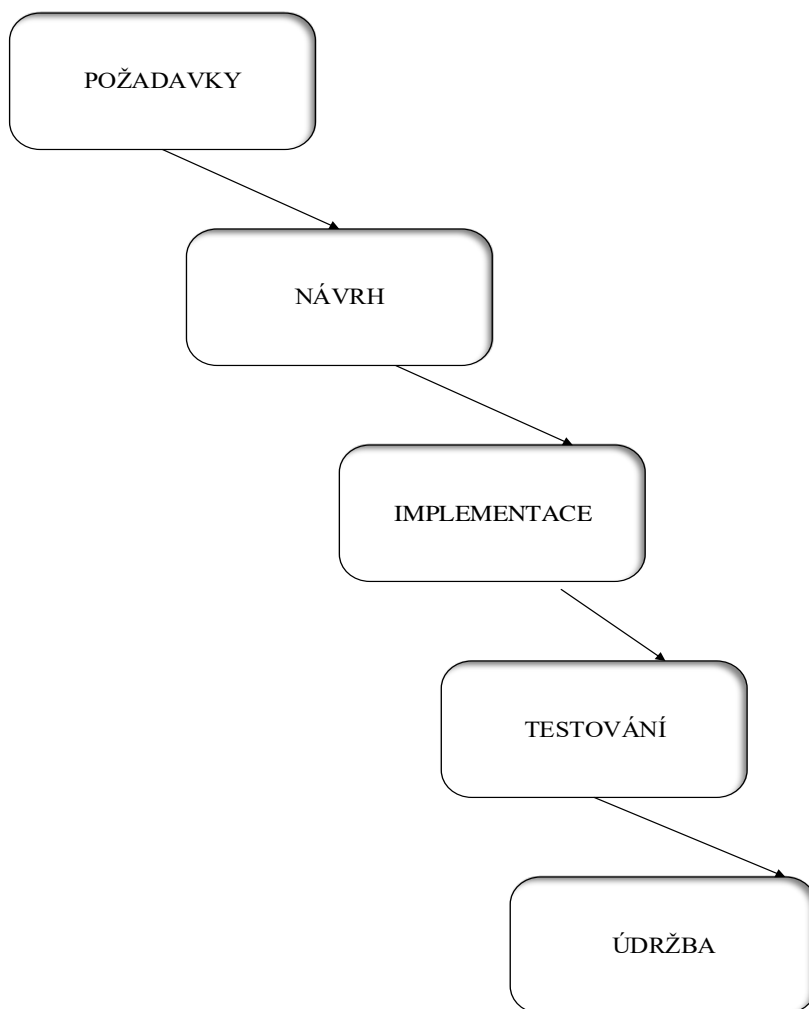
Verifikaci lze chápat jako proces, který má za cíl ověřit, zda dílčí produkt vývoje softwaru odpovídá předem specifikovaným požadavkům. Verifikační aktivity zahrnují následující oblasti:

- Verifikaci samotných požadavků
- Verifikaci návrhu
- Verifikaci zdrojového kódu
- Verifikaci integrace
- Verifikaci dokumentace (8)

Validaci rozumíme potvrzení, že dílčí produkt vývoje softwaru funguje dle očekávání zákazníka, přičemž z jeho pohledu je také daný produkt testován. Problém nalezený během validačního testování indikuje problém s požadavky. Produkt může sice zcela odpovídat specifikacím, ale neodpovídá představě zákazníka. Příkladem může být akceptační testování. Tedy testování na straně zákazníka, kdy se testuje, zda dodaný produkt vykonává to, co zákazník potřebuje. (8)



Obrázek 3: V-model (Zdroj: (8) Vlastní zpracování)



Obrázek 4: Vodopádový model (Zdroj:(8) Vlastní zpracování)

Ve vodopádovém modelu si lze všimnout, že každá fáze následuje až po dokončení fáze předcházející. Pokud však například dojde k chybné interpretaci požadavků, tak se takový problém odhalí až v akceptační fázi testování zákazníkem. Při opravě tak bude potřeba projít znovu celý cyklus a tím dojde k výraznému navýšení nákladů. Naproti tomu **V-model** vyobrazen na **obrázku č.3** řeší tento nedostatek tak, že každý výstup z každé fáze prochází ověřením, což umožní odhalit chyby včas a snížit tak náklady na jejich opravu.

(8)

2.11 Úrovně a základní typy testů

Během vývoje produktu je testování prováděno na různých úrovních. Může tak zahrnovat testování celého systému nebo pouze jednotlivých částí. Podle V-modelu, který byl výše popsán, jsou rozlišovány 4 úrovně testování.

Úrovně testování

1. Testování jednotek (Unit testing)
2. Integrační testování (Integration testing)
3. Systémové testování (System testing)
4. Akceptační testování (Acceptance testing) (8)

2.11.1 Testování jednotek

Jedná se o první fázi testování, kterou provádí převážně sám programátor, který testuje jednotlivé bloky programu. Podle toho, zda se jedná o procedurální či objektově orientovaný programovací jazyk, mohou být jednotlivé jednotky funkce, procedury, metody nebo třídy. Při testování jednotek je cílem testovat každou jednotku nezávisle na ostatních a zjistit, že je jejich chování správné. (8)

2.11.2 Integrační testování

Další fází po testování jednotek je jejich integrace do systému jako celku. Do této fáze testování lze vstoupit, pokud máme k dispozici, alespoň dva moduly, které dohromady poskytují určitou službu. Integrace probíhá tak, že jsou do systému postupně přidávány stále další jednotkově otestované moduly, které musí správně fungovat a zároveň nesmí narušit stabilitu doposud fungujícího systému. Při této fázi testování je cílem nacházet chyby vzniklé spojením a následném vzájemném působení modulů.

Při provádění systémové integrace lze postupovat několika **způsoby**.

- **VELKÝ TŘESK:** Při tomto způsobu jsou všechny moduly integrovány zároveň v jednom kroku. Značná nevýhoda spočívá převážně v tom, že pokud se objeví problém, bude jeho nalezení velice obtížné, neboť se může nacházet kdekoliv.
- **PŘÍSTUP SHORA-DOLŮ:** Pokud je prováděna integrace tímto způsobem, pak integraci začínáme s nejvyšším modulem a postupujeme přidáváním nižších.

- **PŘÍSTUP ZDOLA-NAHORU:** U tohoto způsobu se nejprve integruje nejnižší modul a postupuje se výše. Je zřejmé, že při integraci tímto způsobem nebude systém jako celek fungovat, dokud nebude integrován nejvyšší modul.
- **KOMBINOVANÝ PŘÍSTUP:** Kombinace přístupu shora-dolů a zdola-nahoru na třech úrovních. Na nejvyšší z nich jsou integrovány hlavní moduly pomocí přístupu shora dolů, nejčastěji využívané moduly jsou integrovány zdola-nahoru a ostatní moduly jsou umístěny do prostřední úrovně. (8)

2.11.3 Systémové testování

Po úspěšném integračním testování je systém dostatečně stabilní a připravený pro další fázi testování. Tato fáze ověřuje, zda systém splňuje požadavky specifikované zákazníkem a jde o poslední možnost, jak zachytit a opravit defekty předtím, než by jej objevil zákazník. Tato fáze zahrnuje krom funkčního testování i řadu jiných typů testů. Běžně se provádí následující **testy**:

BEZPEČNOSTNÍ TESTY: Ověřují, zda jsou data chráněna proti neoprávněnému přístupu zvenčí i zevnitř a jsou správná a neporušená.

TESTY ROBUSTNOSTI: Zjišťují, zda systém pracuje s chybnými vstupy, neočekávanými situacemi nebo změnami prostředí.

TESTY POUŽITELNOSTI: Testuje se přehlednost, přímocí, snadné provádění často užívaných funkcí, rozložení prvků uživatelského rozhraní aj. (8)

2.11.4 Akceptační testování

Akceptační testování zákazníkem, respektive uživateli. Smyslem těchto testů je zjistit, zda dodaný produkt splňuje akceptační kritéria, jež jsou stanovena zákazníkem, jako měřitelné a ověřitelné podmínky pro přijetí produktu. (8)

2.12 Smoke a sanity testy

Smoke testování označuje provádění testovacích případů, jejichž účelem je zjistit, zda je systém dostatečně stabilní a všechny jeho hlavní části fungují správně. Smyslem smoke testů není tedy nacházet defekty, ale ověřování připravenosti systému. (8)

Sanity testy se od smoke testů liší jen tím, že testy napříč různými základními funkcemi systému jsou prováděny důkladněji.

Oba zmíněné typy testů spadají pod regresní testování, obecně jsou tyto testy prováděny při různých příležitostech, například před ukázkou systému zákazníkovi. (8)

2.13 SWOT analýza

Jedná se o analytickou metodu zaměřenou na zhodnocení vnitřních a vnějších faktorů ovlivňující úspěšnost organizace. Nejčastěji je tato metoda využita v rámci strategického řízení.

S – strengths-silné stránky

W- Weaknesses-slabé stránky

O-Opportunities-příležitosti

T- threats-hrozby (6)

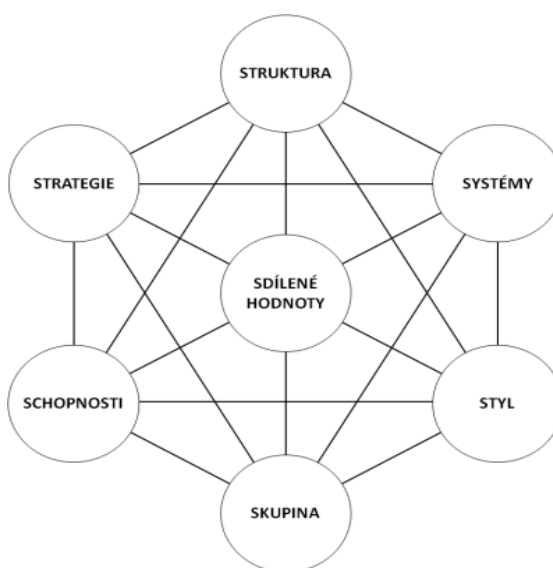
U silných stránek se především posuzuje, jaké výhody má analyzovaná společnost oproti konkurenci, naopak je tomu u slabých stránek.

Příležitosti, tak i hrozby se zaměřují na okolí sledovaného subjektu. Hrozbou pak může být vstup nového konkurenta na trh nebo hrozba v podobě substitutu.

2.14 Analýza vnitřního prostředí

K analýze interních faktorů lze použít analýzu 7S. Ta odhaluje, jakou pozici zaujímá podnik k plnění svých dílčích cílů, zaměřených na sedm klíčových oblastí ovlivňující jeho chod.

Hlavní faktory úspěchu jsou strategie, struktura firmy, spolupracovníci, jejich schopnosti, styl řízení, systémy a sdílené hodnoty firmy. (11)



Obrázek 5: Analýza 7S (Zdroj:(11) Vlastní zpracování)

2.15 Metoda HOS2009

Metoda HOS2009 navazuje na metodu HOS8, která vychází z původní metody HOS. Tato metoda byla vyvinuta na Fakultě Podnikatelské, Vysokého učení technického v Brně docentem Kochem a doktorem Křížem. Navazující metodu HOS2008 poté zpracoval docent Koch a doktor Dovrtěl. Výše zmiňovanou metodu HOS2009 zpracoval a publikoval ve své dizertační práci doktor Neuwirth.

„Metoda nabízí ucelený pohled na informační systém podniku jak v současném stavu, tak i výhledem do budoucna s ohledem na fázi životního cyklu informačního systému firmy a náročnost na informační úroveň firmy“. (9, s.46)

Základním principem hodnocení informačního systému touto metodou je, že za optimální vyvážený informační systém může být považován pouze takový, který nepřináší žádné další zbytečné náklady. Informační systém je tedy vyvážený a zároveň je optimální a stabilní.

Aplikování metody spočívá převážně v dotazníkovém šetření, jednotlivé dotazníky jsou zaměřeny na určité oblasti informačního systému. Celkově se jedná o 10 oblastí a ke každé z nich je sestaveno 10 konkrétních otázek.

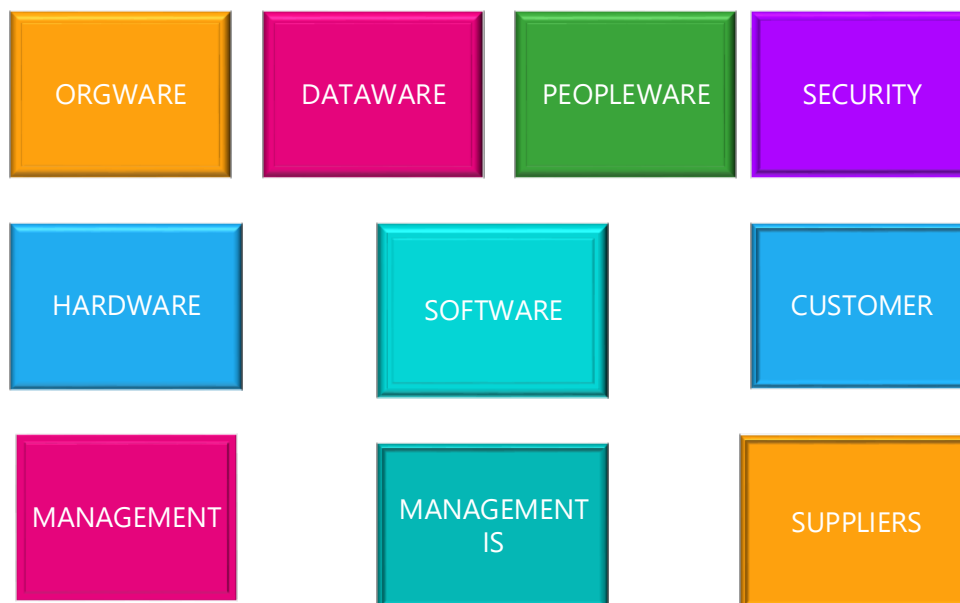
Formulace odpovědí jsou na stupnici Ano, spíše-ano, částečně, spíše-ne a Ne.

Jednotlivé otázky jsou rozděleny do skupin podle kritérií. Ke každému kritériu je přidělena určitá váha podle toho, jak je daná oblast pro firmu důležitá.

Na základě zjištěných hodnot se dále počítají další ukazatelé, kterými jsou např. vztah mezi HW a SW, celkový stav informačního systému a stav jednotlivých oblastí. (9)

2.15.1 Oblasti hodnocení IS metodou HOS2009

Počet oblastí je shodný s počtem dotazníků, jedná se tedy o 10 oblastí, na které se metoda aplikuje.



Obrázek 6: Oblasti hodnocení IS metodou HOS09 (Zdroj:(9) Vlastní zpracování)

Metoda je aplikovaná v několika fázích

1. Před-aplikační fáze
2. Aplikační fáze
3. Fáze tvorby výstupů
4. Fáze zpracování výstupů
5. Fáze interpretace výstupů

Jednotlivé fáze na sebe logicky navazují, není možné některou z nich vypustit nebo se na některou z nich více zaměřit než na jiné. Všechny fáze jsou stejně důležité, je tedy potřeba dodržet předem určené postupy. (9)

2.16 Předaplikační fáze

Tato fáze se dále rozděluje do 3 dílčích částí:

- Seznámení s metodou a jejími omezeními
- Seznámení se s firmou a informačním systémem firmy
- Výběrem vhodných respondentů. (9)

Dva možné přístupy k výběru respondentů:

- Respondentem bude přímo aplikant metody, lze tak hodnocení provést rychleji, hrozí však i značné zkreslení výsledků. Tento způsob tedy není zcela vhodný.

- Aplikant provede průzkum na vybrané otázky u různých respondentů-jde o pomalejší a složitější způsob hodnocení. Tento způsob vede k mnohem přesnějším výsledkům a je doporučován, pokud je to v dané společnosti možné. (9)

2.17 Aplikační fáze

Nejprve se určí optimální a vyvážený stav IS společnosti. Tento stav je určen v závislosti na dvou faktorech:

- Náročnost na informační úroveň společnosti
- Fáze životního cyklu informačního systému

Dále je nutno stanovit váhu jednotlivých odvětví IS, které nabývají hodnot 0.2;0.4;0.6;0.8;1 od nejméně důležitých až po největší váhu daného odvětví.

Stanovení kritérii hodnoty od 1 do 10 od nejméně důležité po vysoce důležité kritérium. Následně lze přejít k samotnému vyplnění dotazníků respondenty. (9)

2.18 Fáze tvorby výstupů

Nyní aplikant převede odpovědi z vyplněných dotazníků na bodové hodnoty pomocí převodové tabulky. Získané hodnoty jsou dosazeny do **vzorce č.1**, ze kterého získáme bodové ohodnocení jednotlivých kritérií. (9)

$$O_i = \sum_{j=1}^m \frac{\sum_{a=1}^n H_{ja}}{n} = \frac{V_j}{\sum_{j=1}^m V_j}$$

Vzorec 1: Výpočet stavu jednotlivých oblastí IS (Zdroj: (9) Vlastní zpracování)

O_i souhrnná hodnota i-té oblasti

V_j váha j-tého kritéria dle důležitosti zkoumaného kritéria pro oblast

m počet kritérií patřících k dané i-té oblasti

H_{ja} a-tá hodnota otázky patřící k j-tému kritériu

n počet kontrolních otázek patřících ke kritériu

Následně stanovíme hodnoty celkového stavu informačního systému společnosti, tak že najdeme minimální hodnotu vyváženosti ze všech měřených oblastí IS. Po zjištění míry

vyváženosti IS společnosti vycházíme z tvrzení „*Informační systém firmy je považován za natolik vyvážený, na kolik je vyvážená jeho nejslabší část.*“ (9, s.61).

2.19 Fáze zpracování výstupů

V této fázi jde o zjištění, zdali je oblast hardwaru v souladu s oblastí softwaru. Slouží k tomu vzorec pro výpočet míry nevyváženosti softwaru a hardwaru.

$$N = \left(\left(\frac{SW}{HW} \right) - 1 \right) * 100$$

Vzorec 2: Výpočet míry nevyváženosti softwaru a hardwaru (Zdroj: Vlastní zpracování (9))

N= celková nevyváženost HW a SW v %

V dalším kroku zjistíme vyváženost IS a jeho komponent. Výsledky pak můžeme porovnat se získaným vyváženým stavem.

Pomocí dalšího vzorce zjistíme míry nevyváženosti jednotlivých sledovaných oblastí IS.

$$N_i = \left(\frac{O_i}{O_{vMin} + V_i(O_{vMax} - O_{vMin})} - 1 \right) * 100$$

Vzorec 3: Výpočet hodnot míry nevyváženosti jednotlivých sledovaných oblastí IS (Zdroj:(9) Vlastní zpracování)

N_i míra nevyváženosti i-té sledované oblasti v %

O_i celková hodnota i-té sledované oblasti

O_{vmin} vyvážená (optimální) hodnota systému – minimální

O_{vmax} vyvážená (optimální) hodnota systému – maximální

V_i význam i-té oblasti IS pro firmu

Následně zjistíme míru nevyváženosti systému výběrem maximální hodnoty ze všech měř nevyváženosti v absolutní hodnotě za jednotlivé odvětví systému. (9)

Zpracování grafických analýz výsledků metody:

- Paprskový graf-zachycuje celkový stav IS. Zobrazuje stavy jednotlivých oblastí, ale i vyvážený, optimální, aktuální i budoucí stav.
- Sloupcový graf-zobrazuje nevyváženost jednotlivých komponent. Můžeme z něj vyčíst i míru celkové nevyváženosti systému.
- Graf technologie-ukazuje vzájemnou vyváženost oblasti softwaru a hardwaru. (9)

2.20 Fáze interpretace výsledků

Na základě získaných dat z grafických výsledků můžeme zhodnotit aktuální stav informačního systému a zvolit vhodnou strategii, která by vedla ke zlepšení celkového stavu a především, ke zlepšení efektivity informačního systému ve sledované společnosti.

Metoda HOS2009 dělí IS na základě měření na čtyři různé skupiny:

- Absolutně vyvážený IS
- Vyvážený IS
- Problémový IS
- Nevyvážený IS

Absolutně vyvážený systém

Provoz takového IS je velmi efektivní. Do žádné oblasti nejsou vynakládány dodatečné náklady a neexistuje žádná oblast, kde by hrozil nedostatek prostředků pro její úspěšný chod. Systém se přizpůsobuje požadavkům uživatelů a umožňuje firmě další rozvoj a nijak ji neomezuje. (9)

Vyvážený systém

Provoz vyváženého systému je pro firmu přínosný. Přesto jsou však do některých oblastí vynakládány nadbytečné náklady, naopak neexistuje oblast, které by se nějak výrazně nedostávalo potřebných prostředků pro její fungování. Systém je přizpůsoben pro potřebu většiny požadavků, které uživatelé od systému vyžadují. (9)

Problémový systém

Provoz takového systému může být již problematický. Do některých oblastí jsou vynakládány nadbytečné náklady a zároveň se některým oblastem nedostává potřebných prostředků. Systém není pro uživatele zcela přizpůsoben podle jejich požadavků. Neumožňuje firmě příliš dalšího rozvoje, firma bývá informačním systémem částečně omezena. (9)

Nevyvážený systém

Pro společnost je tento systém velice neefektivní. Do některých oblastí je investováno nadbytečné množství prostředků na jiné oblasti se pak nedostává. Systém se většinou

nepřizpůsobuje uživatelům a firmě není většinou umožněn další rozvoj. Firma je výrazně omezována informačním systémem. (9)

Hodnocení IS	Technologie	Celkový stav IS	Vyváženost oblastí
Absolutně vyvážený	Poměr mezi HW a SW je vyvážený, pohybuje se mezi -5% až 0%	Celkový stav informačního systému je shodný se stavem optimálním pro firmu v současnosti.	Všechny oblasti informačního systému lze označit za vyvážené, celková nevyváženost max. 0-5%
Vyvážený IS	Poměr mezi HW a SW je vyvážený, pohybuje se mezi -25% až 0%	Optimální stav je zcela pokryt celkovým stavem IS v současnosti	Je povolena nevyváženost u max. 2 oblastí v rozsahu 0-25% nebo i více oblastí v rozsahu 0-15%
Problémový IS	Poměr mezi HW a SW může být max. až 25%	Optimální stav není zcela pokryt celkovým stavem IS v současnosti	Je povolena nevyváženost u max. 2 oblastí vyšší než 25% nebo u 1 oblasti nevyváženost až -25% nebo u 2 oblastí až -15%
Nevyvážený IS	Poměr mezi HW a SW může být libovolný	Optimální stav není pokryt celkovým stavem IS v současnosti	Nevyváženost u více než 2 oblastí vyšší než 25% nebo u 1 oblasti nevyváženost vyšší než -25% nebo u více než 2 oblastí vyšší než -15%

Tabulka 1: Popis souhrnných stavů zkoumaného IS (Zdroj: (9) Vlastní zpracování)

3 Analýza současného stavu

3.1 Základní údaje o společnosti

Obchodní firma: ARTIN, spol. s.r.o.

Sídlo: Brno -Královo Pole, Božetěchova 19,612 00

Identifikační číslo: 255 311 15

Právní forma: Společnost s ručením omezeným

Datum zápisu: 9.6.1998

Základní kapitál: 100000 Kč (13)

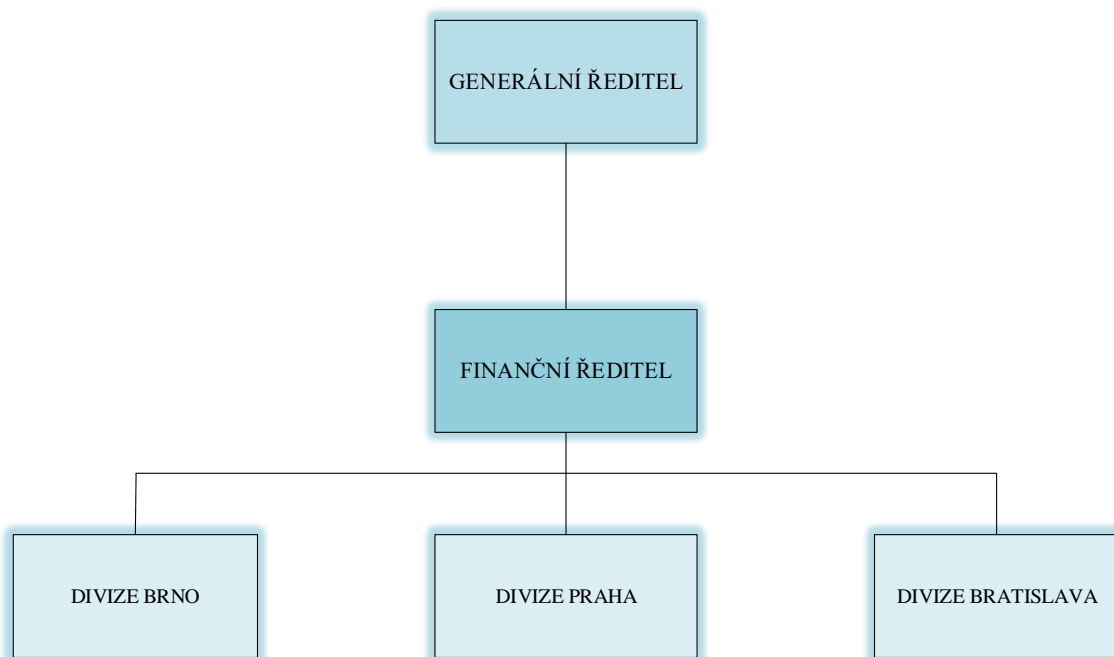
3.2 O společnosti

ARTIN je společnost sídlem v Brně a dalšími pobočkami v Praze, Bratislavě a Košicích. Společnost poskytuje služby především v oblasti vývoje softwaru. Má bohatou klientelu nejen na území České a Slovenské republiky, reference má rovněž i v zahraničí, jako je Rakousko, Švýcarsko nebo Kanada.

Společnost působí především v oblasti telekomunikací, pojišťovnictví, průmyslu a logistice. Jako jedna z mála českých softwarových firem se ARTIN zabývá výzkumem a vývojem v oblasti umělé inteligence. Jako jeden z projektů, který do této oblasti spadá je projekt Roboauta. Do portfolia služeb společnosti spadá i Quality assurance, Support či Konzultace v oblasti IT. (12)

3.2.1 Organizační struktura

Společnost ARTIN pro své činnosti využívá své stálé zaměstnance a několik desítek externistů. V současnosti má kolem 150 zaměstnanců na pobočkách v České a Slovenské republice. Většina těchto pracovníků se zabývá vývojem softwaru. Společnost je rozdělena na divize a pobočky v čele stojí generální ředitel, který je zároveň i společníkem firmy. Představenstvo společnosti tvoří jednatel, který ve firmě vykonává i funkci finančního ředitele a zastřešuje řadu projektů v oblasti umělé inteligence.



Obrázek 7: Organizační struktura (Zdroj: Vlastní zpracování)

3.2.2 Analýza interních faktorů

Strategie

Základní strategií společnosti ARTIN je poskytovat služby v oblasti software v co nejvyšší kvalitě, soustředit se na výsledek, pracovat rychle a co nejjednodušeji.

Těžiště strategie je sdílení informací a s ní ruku v ruce i zpětná vazba. Pro společnost je zásadní, aby poskytované služby byly přínosem pro cílového zákazníka, je tedy důležité, aby zákazník za své vynaložené finanční prostředky získal produkt, který bude mít skutečnou hodnotu, usnadní mu práci, umožní vytvářet vyšší zisky a získat značnou výhodu před konkurencí.

Stejně jako spokojenost zákazníka je pro firmu důležitá spokojenost jejich zaměstnanců. Jejich spokojenost je důležitým faktorem pro budoucí vývoj ARTINu, motto společnosti je **It's in the people**. Společnost podporuje své zaměstnance ve vzdělání, dává jim určitou svobodu v jejich práci, nejsou zde předepsány žádné postupy, jak by měl zaměstnanec postupovat na daném projektu či s jakou technologií by měl přesně pracovat.

Struktura

V čele společnosti stojí generální ředitel spolu s finančním ředitelem, tvoří vedení celé společnosti. Společnost se skládá z několika divizí, poboček a týmů. Organizační struktura je tvořena v hierarchické stromové struktuře. Každý zaměstnanec má právě jednoho svého nadřízeného, bez ohledu na to, zda právě spolupracují na stejném projektu. Neformálně lze společnost rozdělit na personální oddělení, projektové řízení, finance či vývoj, který se skládá z jednotlivých programátorů.

Systém

Pro interní potřeby je využíván systém TimeMisson, který je předmětem zkoumání této práce. Tento systém je vlastním produktem společnosti a slouží především k evidenci pracovní docházky zaměstnanců a umožňuje sledování všech projektů, na kterých firma pracuje. Data z informačního systému slouží, jako zdroj cenných aktuálních reportů a analýz.

Styl řízení

Ve společnosti panuje velice přátelská atmosféra, což značí i velice nízká fluktuace zaměstnanců. Zaměstnanci se navzájem dobře znají, a to i napříč s ostatními pobočkami v České a Slovenské republice. Obecně lze tedy daný styl řízení označit za demokratický.

Spolupracovníci

Vývojový tým je tvořen mladým kolektivem, který se skládá, jak z absolventů různých technických škol, tak i studentů. Řada pracovníků má již několikaleté zkušenosti v oboru. Zaměstnanci nejsou při práci na většině projektů vázáni na určitou technologii, mají tedy prostor neustále rozšiřovat svoje znalosti a učit se nové techniky. Zaměstnanci nepracují pouze na konkrétních zakázkách pro klienty, ale mohou své zkušenosti prohlubovat i v oblasti výzkumu a vývoje. Z pohledu platové politiky společnost kombinuje přístup k hodnocení s ceněním zkušenosti a loajalitu zaměstnanců.

Sdílené hodnoty

Především z výše zmíněné strategie je kladen důraz na vztah se zákazníky, na vytváření přátelské atmosféry na pracovišti, na neustálém vzdělávání zaměstnanců a sdílení informací při práci na společných projektech.

Hlavní hodnoty společnosti

Přínos pro zákazníka

Profesionalita

Dobré místo pro práci

Schopnosti

Nejvýznamnější schopností společnosti je znalost problematiky vývoje softwaru v mnoha oblastech, jako například pojišťovnictví, telekomunikace a průmyslu. V neposlední řadě také bohaté a neustále se rozvíjející se znalosti v oblasti umělé inteligence. Všechny tyto aspekty pomohly společnosti udržet své postavení na trhu i v současném konkurenčním prostředí již řadu let.

3.3 SWOT analýza informačního systému

Silné stránky

Za silnou stránku systému lze považovat jeho vývoj na míru dle potřeb společnosti.

Systém je poměrně široce zaměřen. Od evidence pracovních záznamů jednotlivých zaměstnanců až po sledování veškerých činností spojené s prací na jednotlivých projektech. V jeho prostředí se rovněž tvoří veškeré faktury pro zákazníky společnosti. A jsou zde k dispozici i všechny důležité reporty, jejichž výstupy lze snadno importovat do XLS souborů.

Za další silnou stránku lze považovat zkušené zaměstnance, kteří se podílejí na vývoji produktu a veškeré poruchy či nedostatky jsou vždy odstraněny v co nejkratším možném čase v závislosti na vážnosti konkrétního defektu.

Slabé stránky

Slabé stránky systému spatřuji především v nedostatečném pokrytí systému UNIT testy na serveru, které v současnosti nepokrývají ani polovinu veškeré funkcionality systému. Tyto testy jsou vždy spouštěny před nasazením nové verze systému, aby byly zachyceny případné chyby v kódu. Je tedy potřebné, aby se pokrytí testu rozšířilo, neboť testy, které jsou již vytvořené, nemusejí pokrývat důležité části systému, které jsou nyní pro podnik klíčové.

Další slabou stránku pro správný chod systému mohou být nevyužité automatické testy pro grafické uživatelské rozhraní, které ve společnosti jsou již několik měsíců vytvořeny, ale nebyly využívány ani udržovány. Využitím těchto testů by se mohlo urychlit manuální testování, ale hlavně by se mohly odhalit běžné defekty dříve, než na ně narazí uživatel během každodenní práce se systémem. Další slabinou může být i fakt, že na projektu se často mění pracovníci, je tedy potřeba, aby byly správným způsobem a včas předávány veškeré informace a znalosti související s vývojem systému mezi všechny členy týmu.

Příležitosti

Potencionální příležitostí v rámci systému může být automatizace dalších činností, které systém nyní neumožňuje a které bude potřeba do budoucna do systému implementovat v rámci případného rozšiřování činnosti společnosti.

Příležitost lze vidět i v nových technologiích, které jednak mohou zjednodušit práci na vývoji systému a mohou rovněž přispět k vylepšení grafického rozhraní.

Jak již bylo zmíněno výše, k vyšší kvalitě řízení softwaru mohou přispět i automatické testy.

Hrozby

Poslední zkoumanou oblastí Swot analýzy jsou i možné hrozby, které mohou působit, jak z okolí firmy ve formě legislativních změn a konkurence, tak i hrozeb přímo uvnitř společnosti. Mezi možné hrozby analyzovaného systému lze zařadit časté a dlouhé výpadky systému, které mohou způsobit ztrátu důležitých dat a informací, které tak v danou chvíli nejsou dostupné.



Obrázek 8: SWOT analýza informačního systému (Zdroj: Vlastní zpracování)

3.4 Servery

Brněnské pobočky mají celkem 8 fyzických serverů. Na 4 z těchto serverů běží systém Esxi, který umožňuje nasazení a běh virtuálních zařízení, jako je například web server nebo builder Jenkins. Na dalších dvou serverech jsou pomocí síťového protokolu iSCSI připojena datová pole pro systém ESXi. Nevýhodou systému ESXi je, že způsobuje občasné odpojení těchto datových polí a tím i výpadky aplikací, které na těchto serverech běží. Tyto výpadky jsou zpravidla řešeny v co nejkratším čase, opravou databáze. V občasných případech, však nelze databáze opravit a je nutno data obnovit z poslední zálohy. V takovýchto případech, je pak možné, že uživatelé ztratí část své práce, která nebyla poslední zálohou zachycena.

Na dalších serverech jsou pak instalovány aplikace, které mají na starosti monitoring sítě a zálohování. (24)

Web server	Zential	GITLab
Proxy server	VPN	Jenkins
TimeMission	SVN	Jira

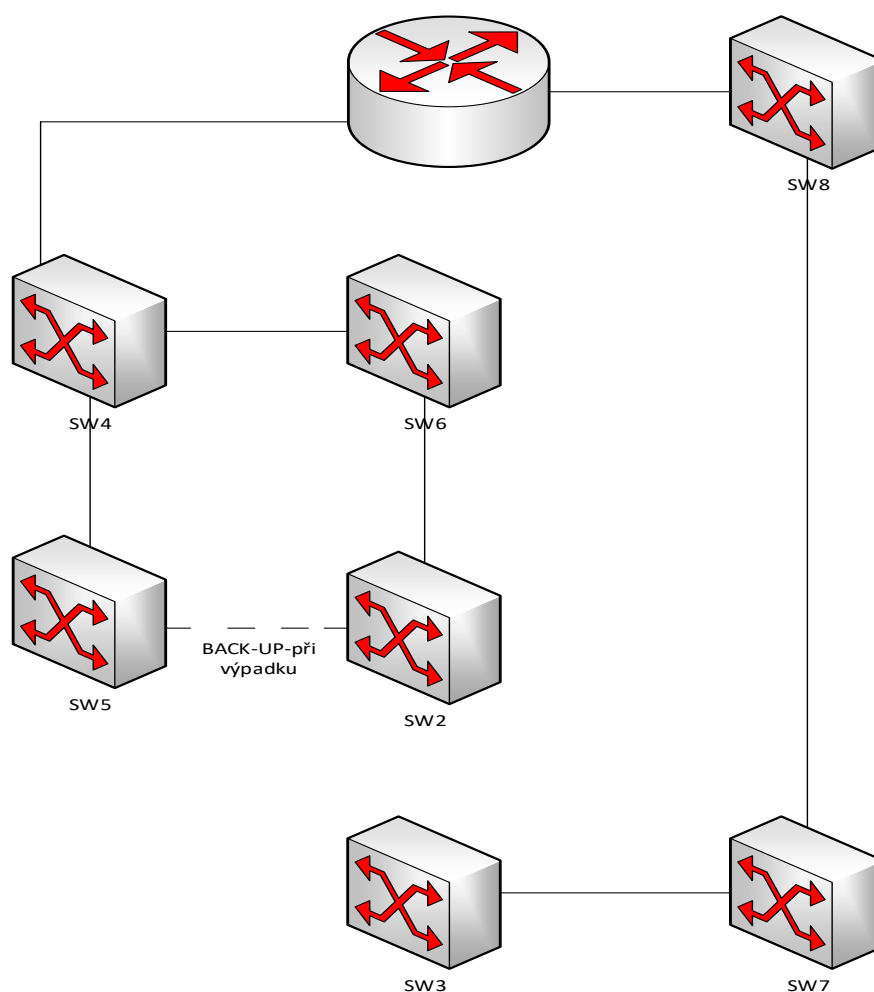
Tabulka 2: Ukázka aplikací virtuálních serverů (Zdroj: Vlastní zpracování)

3.5 Síť

Důležitým prvkem pro firmu je bezesporu počítačová síť. Všechny pracovní stanice na pobočkách jsou síťově propojeny. Stejně tak i veškeré tisková zařízení mají síťové rozhraní a jsou kabelově zapojeny do LAN. Díky síťovému propojení je možné sdílet soubory z jednoho počítače na dalších stanicích a tisknout lze téměř z každého počítače, pokud jsou nainstalovány potřebné ovladače. Brněnské pobočky jsou mezi sebou propojeny IPSec tunelem, jedná se o rozšíření IP protokolu založené na autentizaci a šifrování každého IP datagramu, který projde přes síť. IPSec ověřuje původ dat, jejich integritu, důvěrnost a ochranu proti útoku replay attack, nebo-li útoku přehráním, který umožní útočnickovi zachytit a pozměnit data, která přes síť prochází. (24)

3.6 Logické zapojení a topologie sítě

Grafické znázornění zapojení sítě je zobrazeno na **obrázku č.9**. Zapojení je následovné k routeru jsou připojeny hlavní páteřní switche, které jsou zapojeny do kruhu, z kterého jsou vyvedeny odbočky na koncové porty pro uživatele. Na tyto switche jsou umístěny wifi antény, které zaměstnancům umožňují kromě kabelového připojení do sítě využít i bezdrátové připojení. Kruh je spravován pomocí Rapid Spanning Tree protokolu. Tento protokol zajišťuje stabilní připojení síťovým mostem mezi jeho porty a individuálními LAN sítěmi, které jsou k těmto portům připojené. Další výhodou je že, v případě poruchy či odebrání nebo přidání síťového zařízení nevznikají dočasné smyčky v topologii sítě.



Obrázek 9: Schéma zapojení počítačové sítě (Zdroj: Vlastní zpracování)

3.7 Zálohování aplikačních dat a serverů

Ve firmě se provádí hned několik záloh a to zálohy virtuálních serverů, dat a celých databází.

Zálohy z virtuálních serverů jsou ukládány v časových intervalech, den, týden, 2 týdny, 3 týdny a měsíc. Aplikační data se zálohují pomocí externích disků a to vždy jednou týdně, každý týden se tedy vždy v serverovně manuálně vymění externí disk a externí disk se zálohou, je pověřenou osobou odnesen mimo společnost. V případě jakékoliv havárie lze tedy z těchto disků, co nejrychleji obnovit infrastrukturu společnosti. I

v případech výpadku elektrického proudu, je celá kritická infrastruktura vybavena záložním zdrojem UPS, který pokryje zhruba 20 minut výpadku elektřiny. (24)

3.8 Monitorování sítě, hardwaru a služeb

K monitorování sítě, síťových zařízení a hardware využívá společnosti systém Zabbix. Tento systém podporuje platformy Linux, Windows a UNIX. Pomocí Zabbixu jsou monitorovány jak servery, tak lze s jeho pomocí sledovat dostupnost a citlivost standardních služeb a to bez nutnosti instalace softwaru na monitorovacích zařízeních. Monitorovány jsou i síťová zařízení router a switche, pomocí systému lze sledovat využití sítě, procesoru, paměti a stavu portů. U hardware se především monitoruje teplota, rychlost ventilátoru, napětí a stav disku. (24)

3.9 Analyzovaný systém Time Mission

Systém Time Misson je vlastním produktem společnosti. Systém slouží především k evidenci pracovní doby zaměstnanců a sledování všech projektů, na kterých firma pracuje. Data z informačního systému slouží především vedení, jako zdroj cenných aktuálních reportů a analýz. V systému lze sledovat ziskovost jednotlivých projektů, podklady pro vyúčtování provedených služeb klientům nebo také podklady pro mzdovou agendu.

Koncept aplikace

Koncept aplikace podporuje stromovou strukturu pro oblasti organizační struktury, uživatelské struktury a struktury projektu.

Organizační struktura: Společnost je definována divizí, oddělením a týmem.

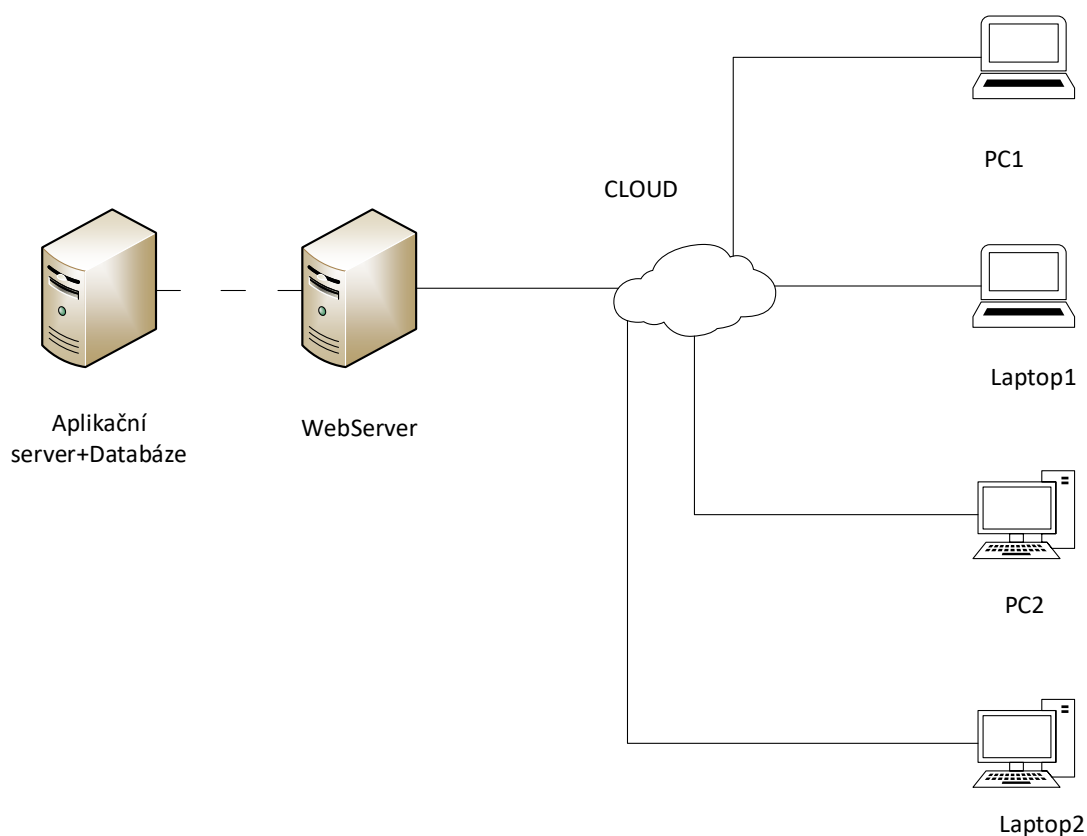
Uživatelská struktura: Každý uživatel má jednoho nadřízeného.

Struktura projektu: Nadřazenému projektu podléhají všechny sub-projekty.

3.9.1 Architektura a přístup do systému

Time Misson je webová aplikace. Přístup do systému je tedy zajištěn pomocí webového prohlížeče na klientském počítači, není tedy potřeba nic instalovat, systém lze využívat, na jakémkoliv počítači s přístupem k síti bez ohledu na operační systém. Poskytovatel

softwaru v tomto případě tedy ARTIN má svůj web server, který komunikuje s jádrem informačního systému. Webserver dynamicky generuje HTML stránky. Koncový uživatel pak tyto stránky zobrazuje s využitím svého webového prohlížeče na klientském počítači. Dnes se již u většiny modulů využívá technologie AngularJS, který generuje HTML stránku přímo v prohlížeči uživatele a server pouze poskytuje data.



Obrázek 10: WebServer architektura (Zdroj: Vlastní zpracování)

3.9.2 Oprávnění

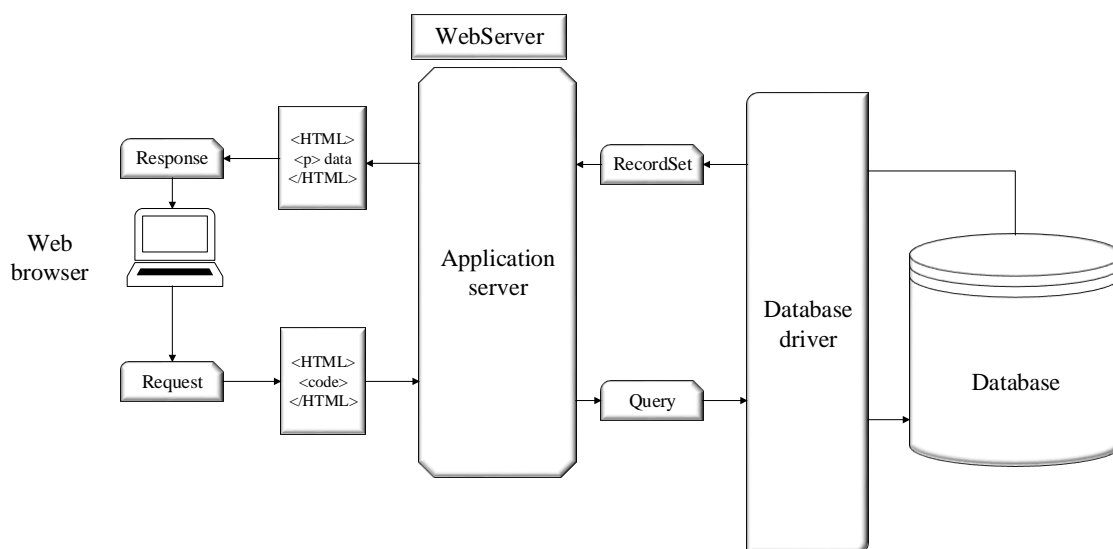
Každý uživatel, který má oprávnění se systémem pracovat vlastní přihlašovací údaje. Přístup ke konkrétním datům v systému se odvíjí od uživatelských rolí, které jsou již definovány v systému. Každý uživatel má tedy přístup pouze k některým datům. Výjimkou je pak administrátor, který má přístup ke všem částem systému.

3.10 Serverové skriptovací technologie pro dynamické generování HTML stránek využívané v Time Mission

Jak již bylo zmíněno výše, Time Mission je webová aplikace. Webovou aplikaci lze chápat jako kolekci statických a dynamických webových stránek, kde statická webová stránka zůstává neměnná. Když na takovou stránku zadá uživatel požadavek, webový server odešle stránku webovému prohlížeči beze změny. Naopak dynamickou webovou stránku server modifikuje před odesláním webovému prohlížeči, který o ni žádal. Vývojáři k tomuto účelu využívají serverové technologie JSP a AngularJS. (23)

JavaServerPages

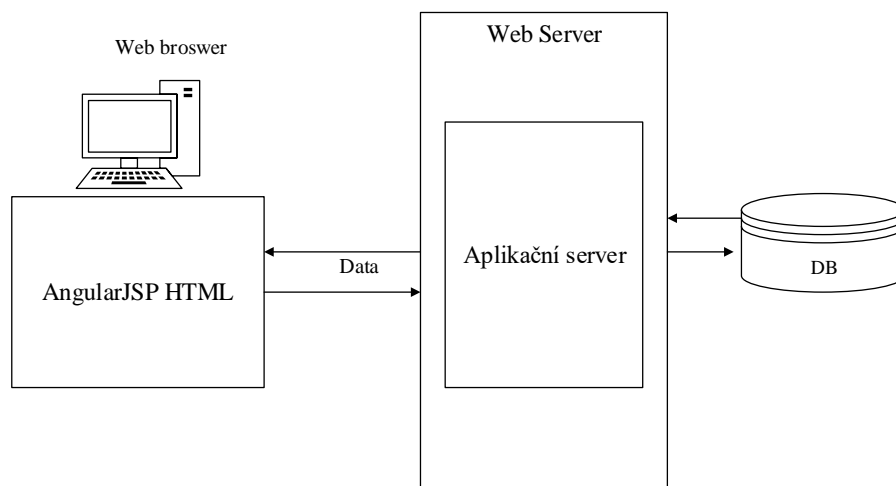
HTML stránka se vytváří na serveru a potom se již hotové HTML pošle do prohlížeče uživatele, server obstarává kompilaci HTML.



Obrázek 11: Skriptovací serverové technologie JSP(Zdroj: Vlastní zpracování)

AngularJS

HTML stránka se vytváří v prohlížeči uživatele. Uživatel si načte klientskou angularJS aplikaci a ta už potom sama stahuje ze serveru data a kompiluje HTML stránky v prohlížeči, server pouze poskytuje data.



Obrázek 12: Skriptovací technologie AngularJS (Zdroj: Vlastní zpracování)

3.11 Moduly systému

Time Mission je vysoce modulární systém, poskytující svou funkcionalitou široké množství agend společnosti. V této části budou jednotlivé moduly blíže popsány.

PRACOVNÍ ZÁZNAMY: Jeden z klíčových modulů systému. Modul, ve kterém si zapisuje zaměstnanec svoji docházku. Nežadává se zde pouze čas, ale také projekt, na kterém zaměstnanec pracoval a typ práce, kterou na tomto projektu vykonával. Na základě těchto atributů, lze posléze sledovat náklady, které se váží k danému projektu a typu práce, současně se tyto data využívají, jako podklad ke mzdové agendě.

NÁKLADY: Další oblastí systému je modul sloužící k vytváření a evidenci nákladů. V tomto modulu si lze snadno zjistit, jaký typ nákladu je přiřazen k určitému projektu za vybrané období.

ALOKACE: Modul umožňuje přidělovat alokace uživatelům na vybrané projekty. Tyto údaje slouží řídicím pracovníkům a personal leaderům, jako přehled, ze kterého lze zjistit, na jakém projektu a v jakém časovém období bude pracovník nedostupný pro případnou práci na jiném projektu nebo naopak, který pracovník má nedostatečnou alokaci a lze mu přidělit práci i na dalších projektech.

V rámci stromové hierarchie může alokace přidělovat pouze nadřízený všem svým podřízeným pracovníkům nebo administrátor libovolnému uživateli.

OBJEDNÁVKY: Evidence objednávek klientů. Uživatel si v této části může zjistit veškeré objednávky vážící se ke konkrétnímu projektu, dobu trvání objednávky, cenu či informaci, zda byla objednávka již fakturována.

FAKTURACE: Time Misson umožňuje tvorbu a přehled veškerých faktur, které jsou klientům fakturovány za poskytované služby.

TISKY: Modul umožňující získat souhrnné informace za určité období ze všech oblastí informačního systému. Tyto reporty pak slouží uživatelům, jako případný nástroj pro rozhodování. Většina těchto reportů umožňuje i import dat do XLS souborů.

SPRÁVA: Možnost oprávněného uživatele upravit pracovní záznamy, náklady, benefity nebo alokace svým podřízeným. Ke změně jsou nabízeny tedy pouze ty záznamy, které se váží na uživatele, kteří jsou v hierarchické struktuře podřízeni aktuálně přihlášenému uživateli.

SCHVÁLENÍ: Slouží ke schválení projektů oprávněným uživatelem.

CRM: Modul řízení vztahu se zákazníky je zaměřen na vztah mezi firmou a jejími klienty. Systém TimeMisson umožňuje v rámci tohoto modulu vytvářet marketingové akce nebo si vést přehled o možných příležitostech na nové projekty, a v jakém stavu se daná příležitost nachází. Takto získaná statistická data, lze později využít k vyhledávání nových obchodních příležitostí.

EDM a Workflow

Tato část systému je zaměřena na správu dokumentů mezi firmou, odběrateli a zaměstnanci. Veškerá dokumentace je na jednom uložišti v databázi. Dokumentace je pořizována v různých formátech a pochází z více různých procesů ve firmě, které se dále třídí podle zdroje, ze kterého pochází. Jsou zde všechny příchozí i odchozí faktury. Evidují se zde elektronické i naskenované dokumenty. Schvalování samotných dokumentů je definováno na dvou úrovních. Proces schvalování **1. úrovně** a proces schvalování **2. úrovně**. Při tvorbě dokumentu je defaultně před-vyplněné povinné pole schválení na úrovni 2. Schvalování dokumentu na 1. úrovni není povinné pole. Pokud jsou však obě úrovně u dokumentu vyplněné, postupuje se následovně.

STAV PŘI ODMÍTNUTÍ SCHVÁLENÍ: Tento případ může nastat, pokud zadavatel dokumentu například zaměstnanec, vytvoří dokument a vyplní dokument chybně či neúplně. Proběhne-li odmítnutí schválení na **1. úrovni** a na **2. úrovni** je dokument schválen, je odeslán email tvůrci dokumentu a po odstranění nedostatků, bude dokument

opět předložen ke schválení. Pokud proběhne odmítnutí schválení na **2. úrovni** a dokument je schválen na **1. úrovni**, je odeslán email schvalovateli na

1. úrovni s uvedeným důvodem, proč nebyl dokument schválen **schvalovatelem na**

2. úrovni. Pokud proběhne odmítnutí schválení na obou úrovních je informován opět tvůrce dokumentu a dokument není schválen.

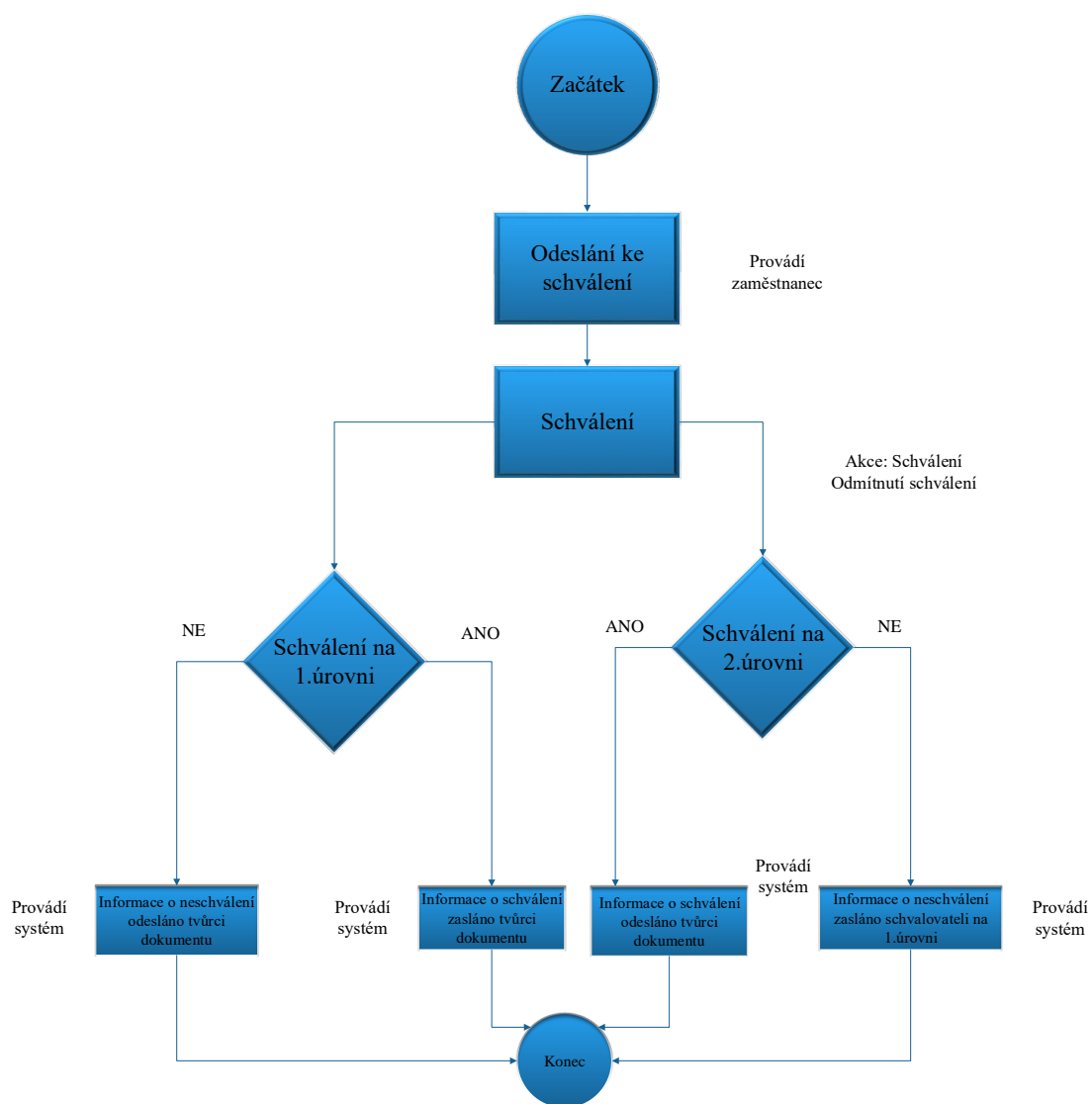
STAV PŘI SCHVÁLENÍ

Dokument je schválen, pokud je schválen na obou úrovních nebo v případě, že je schválen na 2. úrovni a 1. úroveň schválení nebyla, při tvorbě dokumentu nikomu přidělena.

SCHVALOVACÍ ROLE

1.ÚROVEŇ: je vždy vlastník projektu

2. ÚROVEŇ: je vždy schvalovatel pobočky



Obrázek 13: Vývojový diagram schvalovacího procesu dokumentu. (Zdroj: Vlastní zpracování)

HR: Modul sloužící především personálnímu oddělení. HR pracovník si do systému zapisuje pohovory s možnými budoucími pracovníky. Každý personalista, tak má přehled o svých pohovorech i pohovorech ostatních HR pracovníků ze všech poboček. Evidují se zde i interní pohovory se zaměstnanci. Pravidelně jsou pracovníci personálního oddělení informováni prostřednictvím notifikačních emailových zpráv s informacemi o blížících se akcích, pracovních pohovorech nebo důležitých interních datech jako např. konce zkušebních dob jednotlivých pracovníků.

Personalista, tak nemusí pravidelně tato upozornění sledovat přímo v systému, ale je informován i prostřednictvím firemního emailu.

Modul rovněž shromažďuje informace o jednotlivých inventárních položkách, jakému zaměstnanci a kdy, byly přiděleny jednotlivé položky, jedná se především o elektroniku, která byla zaměstnancům propůjčena k jejich práci.

KONFIGURACE: Umožňuje konfigurovat různé části systému.

NÁSTROJE: Slouží k hromadným operacím v Time Misson, jako hromadné přidělení benefičních bodů zaměstnancům, hromadné změny záznamů, přepočítání cen u pracovníků nebo slučování klientů, které slouží k odstranění duplicit do jednoho záznamu.

- Přepočítání cen
- Hromadné přidělení benefičních bodů
- Hromadné změny záznamů
- Slučování klientů



Obrázek 14: Jednotlivé moduly systému Time Misson (Zdroj: Vlastní zpracování)

3.12 Issue Tracking Systém

V rámci zajišťování a sledování kvality vývoje softwaru se ve společnosti využívá takzvaný Issue Tracking systém, který umožňuje evidovat chyby v aplikacích, které jsou objeveny buď koncovými uživateli, nebo vývojáři. Aplikace Issue Tracking systém neumožňuje pouze evidování chyb, jak je tomu u aplikací Bug Tracking Systém, ale umožňuje evidovat i návrhy na vylepšení či požadavky na nové funkce aplikace. Ve firmě je k tomuto účelu využívány nástroj JIRA software.

3.12.1 JIRA software v6.3.6

JIRA je software, který monitoruje a eviduje úlohy týkající se činností spojených s realizací projektů převážně v oblasti IT technologií. Poskytuje přehled o stavu projektu. Prostřednictvím této aplikace lze nastavit priority určitým úkolům a přiřadit je ke zpracování konkrétním pracovníkům. V současné době se JIRA využívá mimo jiné i na analyzovaný systém Time Mission. Uživatelé zde nahlašují defekty systému, na které narazili během své práce se systémem.

Lze vytvořit několik druhů Issues:

BUG

IMPROVEMENT

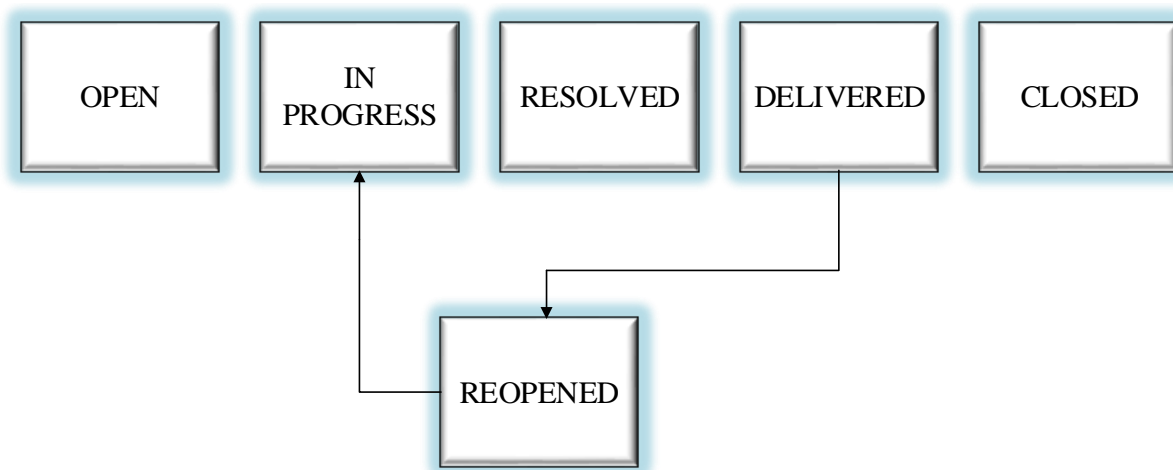
TASK

NEW FEATURE

3.12.2 Životní cyklus ohlášeného ISSUE

Při zaznamenání nového issue je důležité, aby bylo uživatelem, který dané issue reportuje správně popsáno, kde se chyba v systému přímo nalézá, jakou prioritu tento problém má. Na základě zvolené priority je pak stanoveno, jak rychle se musí daný problém vyřešit a v neposlední řadě uvést, jakému vývojáři bude přiřazen. Vytvořené issues mají dále různé stavy, ve kterých se nachází. Nově vytvořené jsou ve stavu OPEN, kde čekají na zpracování osobou, které byly přiřazeny. Pokud je issue ve zpracování vývojářem změni se stav na INPROGRESS. Jakmile je opraveno, je stav změněn na RESOLVED, kdy bylo dané issue opraveno přímo v kódu, avšak ještě nebyla změna nasazena na produkci tedy na ostrou verzi systému. Uživatel tedy změnu v systému nevidí. V této fázi se čeká, až pověřená osoba tyto opravy nasadí na produkci. Nejdříve se nasadí opravené issue na

testovací verzi systému a přechází do stavu DELIVERED. Jakmile je tato oprava nasazena na testovací verzi systému, přichází na řadu testování. Pokud je tato oprava v pořádku, je již nasazena na produkci a issue je ukončeno, je tedy ve stavu CLOSED. Pokud problém setrvává, vrátí se opět na opravu vývojářům a je označeno jako REOPENED.



Obrázek 15: Životní cyklus ISSUE v JIRA software (Zdroj: Vlastní zpracování)

3.13 Interní systém znalostí

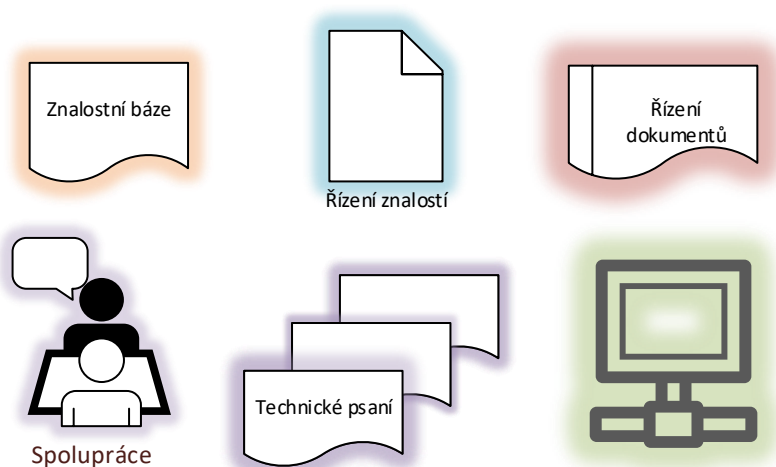
V softwarové společnosti existuje celá řada procesů, které se s různou četností opakují. Ne všechny takovéto procesy lze automatizovat pro drobné nuance nebo překážky. Většinou tak odpovědný pracovník stráví značnou část řešení takových problémů zkoumáním a vyhledáváním informací a postupů. Další rozměr získává tato skutečnost při vstupu dalších pracovníků s podobným problémem. Je tedy potřeba zajistit prostředek pro uchování těchto znalostí a jejich další sdílení. K takovému účelu slouží interní systém znalostí.

3.13.1 Confluence

Confluence je ve firmě využíván, jak je výše popsáno, jako interní systém znalostí. Slouží k firemní komunikaci a sdílení informací. V aplikaci lze komunikaci rozdělovat do jednotlivých vláken dle týmu, projektu nebo dle tématu. Uživatel se pak snadněji orientuje v celé aplikaci a může využít i fulltextové vyhledávání.

Aplikace poskytuje využití předpřipravených šablon s velice jednoduchou editací jednotlivých šablon. Šablony se nejčastěji tvoří pro produktové požadavky, seznamy různých souborů, články, technické dokumentace nebo návody. Informace jsou sdíleny napříč všemi pobočkami společnosti jak v České republice, tak i na Slovensku. Velkou výhodou v této aplikaci spatřuji převážně v tom, že umožňuje informace shromažďovat na jedno místo, což by například pouhá emailová komunikace nedokázala zajistit, informace tak nejsou rozptýleny v emailových schránkách.

Přístup je zajištěn opět pomocí webového prohlížeče. K aplikaci tedy lze přistupovat odkudkoliv, kde má uživatel přístup k síti, je však potřebné se do aplikace autentizovat pomocí přihlašovacích údajů.



Obrázek 16: Možnosti využití aplikace Confluence (Zdroj:(16) Vlastní zpracování)

3.14 Version control systém (VCS)

Verzovací kontrolní systémy jsou používány ke správě zdrojového kódu aplikací. Mezi nejznámější verzovací systémy lze řadit SVN, Git nebo Microsoft Team Foundation Server. Většina těchto nástrojů je nasazována na server. Primárním účelem využití je zaznamenávání změn při vývoji projektu a jejich propagace napříč celým týmem a mezi různými prostředími. (14)

Mezi hlavní výhody VCS lze řadit:

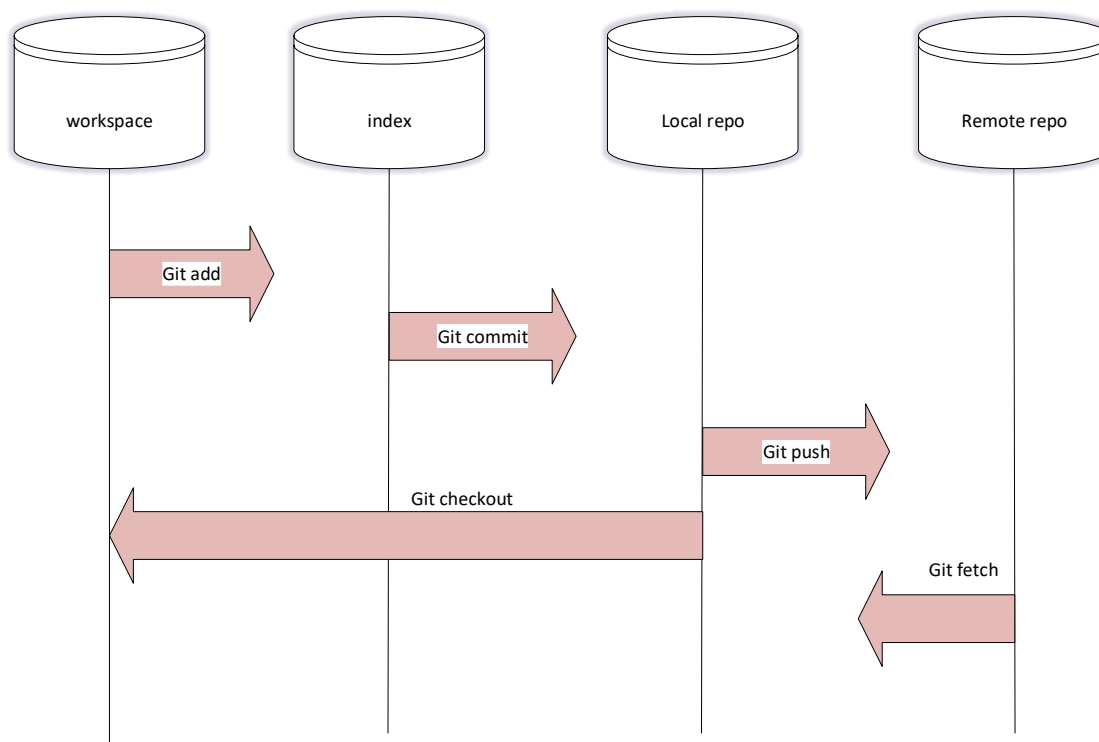
- **Spolupráce:** spolupráce celého týmu, jehož členové nemusejí být na stejném místě a mohou být i rozdílného profesního zaměření (vývojář, tester, vedení)

- **Bezpečné nasazování aktualizací:** Možnost procházet historii změn, je umožněn návrat k předchozí verzi, pokud se nová verze zdá jako nevhodná.
- **Dokumentace:** Záznamy z verzovacího systému tvoří vývojářskou dokumentaci a přidávají i časový rozměr, díky kterému lze zpětně sledovat velikost vývoje.
- **Change log:** evidence záznamů veškerých změn. (14)

3.14.1 GitLab

Ke správě verzí je ve společnosti využívány webový git repozitář GitLab. Tento nástroj je využíván i u projektu Time Mission, kde vývojáři ukládají všechny své verze související s daným projektem. Tedy jakékoliv změny v kódu, které provedly. V GitLabu lze sledovat, kdo vytvořil, jakou část kódu a co se konkrétně změnilo.

Aplikace umožňuje větvení souborů, díky kterému si vývojáři mohou vytvářet nové funkcionality, aniž by tím ovlivnili celý projekt. Jestliže pak tyto změny v kódu nechtějí využít, nemusí tyto změny zahrnovat do projektu a vrátí se na původní místo v historii před psáním nefunkční či jinak špatné navržené funkcionality. V následujícím obrázku jsou zobrazeny dílčí úkony, které probíhají se souborem během ukládání změn do repozitory v GitLabu.



Obrázek 17: Proces ukládání souborů do repozitáře v GitLab (Zdroj: Vlastní zpracování)

Workspace je složka se soubory, se kterou uživatel pracuje, kterou má uložen na svém klientském počítači. Veškeré soubory v této složce nemusí být uloženy i v repozitáři. Soubory, které budou uloženy do repozitáře jsou pak ve složce index nebo také staging area, jedná se o virtuální složku se soubory, které budou během commitu přeneseny do lokálního repozitáře. Ve fázi push jsou pak složky přeneseny do vzdáleného repozitáře, který běží na serveru. Dále pak probíhá synchronizace mezi obsahem lokálního a vzdáleného repozitáře.

K repozitáři lze přistupovat pomocí příkazové řádky, to však od uživatele vyžaduje určité znalosti pro práci s tímto nástrojem. Jednodušší řešení pak může být ve využití grafického rozhraní, například v IDEA IntelliJ nebo Atlassian source Tree.

3.15 Continuous Integration

Continuous integration (CI) proces, který umožňuje společnosti urychlit práci na vývoji softwaru, na rychlejší dodání nových verzí a na snížení chybovosti analyzovaného

systemu, a to vše díky automatizaci více úkonů, které jsou potřebné při vývoji, testování a nasazování aplikace.

Ve společnosti je pro účely continuous integration využíván build server JENKINS.

3.15.1 JENKINS

Jenkins je open source automatizační server poskytující stovky pluginů, které podporují nasazení a automatizaci jakéhokoliv projektu.

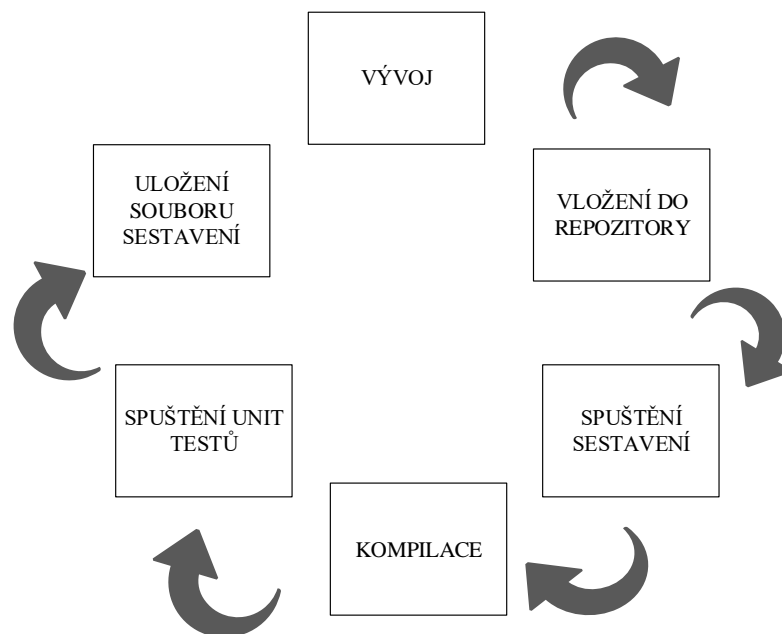
Při tvorbě nové verze systému, umožní tento nástroj celý proces buildu automatizovat. Automatizace **buildu** na projektu Time Misson vypadá pak následovně.

Soubor s upraveným projektem je vložen do repozitáře, následně na to je spuštěno sestavení, po kterém následuje kompilace kódu. Po úspěšné kompilaci jsou spuštěny Unit testy, pokud testy procházejí, proběhne uložení souborů sestavení. Výsledkem buildu je pak soubor, který může být nasazen na testovací prostředí nebo na produkci. Ne každý build je však nasazován.

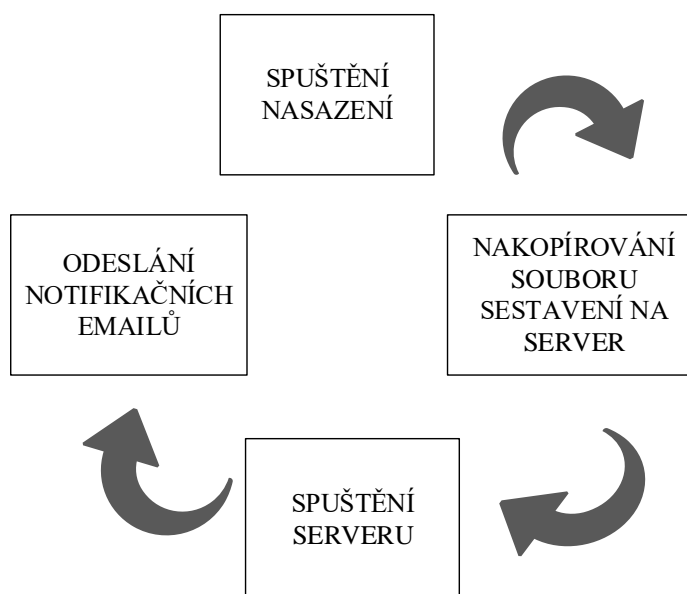
Dalším procesem je pak proces nasazení buildu, který se skládá z následujících kroků. Spustí se nasazení nové verze, soubory sestavení jsou nakopírovány na server, posléze na to se spustí server a vybraným osobám, je zaslán notifikační email o nasazení nové verze.

Shrnutí

- Licence: MIT
 - Platforma: Java
-
- + Množství přehledně zobrazených informací
 - + Možnosti notifikace
 - + Rozšiřitelnost
 - + Zdarma
 - + Jazyková podpora
 - + Jednoduchá tvorba workflow
 - Složitá nastavení některých funkcí
 - Zastaralé uživatelské rozhraní



Obrázek 18: Proces buildu (Zdroj: Vlastní zpracování)



Obrázek 19: Proces nasazení (Zdroj: Vlastní zpracování)

3.16 Metoda HOS2009 hodnocení optimality a vyváženosti IS.

V této části práce bude aplikovaná metoda HOS09, která byla pospána již v teoretické části této diplomové práce. Postup aplikace bude tedy spočívat ve zpracování několika fází.

1. Předaplikační fáze
2. Aplikační fáze
3. Fáze tvorby výstupů
4. Fáze zpracování výstupů
5. Fáze interpretace výstupů

3.16.1 Předaplikační fáze

Předaplikační fáze se skládá ze tří částí, první z nich je seznámení se s metodou a jejími omezeními, seznámení s firmou a informačním systémem firmy a poslední částí je pak výběr vhodných respondentů.

Seznámení se s metodou, jejími omezeními a informačním systémem firmy

Pro bližší pochopení problematiky jsem čerpala z fakultních zdrojů a samotné disertační práce, vztahující se k dané problematice. Seznámení se systémem pak probíhalo v rámci mé pracovní činnosti ve firmě na pozici testera analyzovaného systému.

Výběr vhodných respondentů:

Jednotliví respondenti byli vybíráni na základě zkoumané oblasti. Systémová oblast byla zodpovězena správci informačního systému Time Mission. Oblast customers pak koncovými uživateli, tedy samotní zaměstnanci firmy. Na oblast HW a SW odpovídali lidé z oddělení support IT a oblasti MA a MIS pak majitelé a vedení společnosti. V **tabulce č. 3**. Můžete vidět počet respondentů v jednotlivých oblastech.

Pohled	Oblast	Počet respondentů
Systémový pohled	OW	4
	PW	4
	DW	4
	SEC	4
Technologický pohled	HW	2
	SW	2
Pohled Managementu	MA	2
	MIS	2
Koncový uživatelé	CUS	26
Okolí společnosti	SUP	1

Tabulka 3: Přehled počtu respondentů v jednotlivých oblastech (Zdroj: Vlastní zpracování)

3.16.2 Aplikační fáze

Tato fáze se opět skládá z několika částí. Nejprve se je potřeba stanovit celkový vyvážený optimální stav IS firmy, **určit váhy** jednotlivých oblastí IS, **stanovit význam** jednotlivých oblastí IS pro firmu a v poslední části se pak přechází k samotnému **zodpovězení kontrolních otázek** na jednotlivé oblasti vybranými respondenty.

Stanovení celkového vyváženého, optimálního stavu IS firmy

Celkový optimální a vyvážený stav informačního systému zjistíme:

- Stanovením náročnosti firmy na informační systém: náročnost je stanovena, na základě náročnosti firmy na informační systém a motivaci firmy k používání informačního systému.
- Fáze životního cyklu informačního systému

Náročnost si určuje aplikant metody sám. Neexistuje totiž obecný faktor, který by tuto náročnost určoval, může se jednat o faktory, jako například obor, ve kterém firma působí, cíle kterých chce dosáhnout nebo přínosy, které do budoucna očekává. Náročnost firmy na systém se stanoví pomocí ohodnocení následujících faktorů na stupnici 0–1, kde **0** označuje **nedůležitý faktor** a **1** potom **faktor velice důležitý**.

Hlavní faktory ovlivňující náročnost na informační systém firmy:

1) Náročnost na rychlost, efektivnost vnitropodnikové komunikace.

- 2) Náročnost na rychlost, efektivnost komunikace firmy s okolím.
- 3) Náročnost na kvalitu, aktuálnost, validitu, kvantitu informací v IS.
- 4) Náročnost na bezporuchový chod IS.
- 5) Náročnost na obnovu IS po poruše.
- 6) Náročnost na rychlou použitelnost IS. (9)

Náročnost na informační systém firmy	Průměrně hodnocení faktorů
Nízká	< 0,4
Střední	0,4-0,6
Vyšší	0,6-0,8
Vysoká	>0,8

Tabulka 4: Náročnost na IS a jeho hodnocení (Zdroj:(9) Vlastní zpracování)

Faktor	1	2	3	4	5	6	Průměr
Hodnota	0,8	0,5	0,8	0,9	0,8	0,7	0,75

Tabulka 5: Náročnost na IS firmy (Zdroj: Vlastní zpracování)

Náročnost firmy na informační systém je dle **tabulky č. 5** vyšší.

Motivace firmy k používání informačního systému

Tato část analýzy se zabývá motivy firmy, které vedou k používání IS při různých podnikových aktivitách. Výsledná hodnota se stanoví opět na základě ohodnocení jednotlivých faktorů na stupnici **0–1**, kde **0 označuje nedůležitý faktor a 1 faktor vysoce důležitý**.

Faktory ovlivňující motivaci firmy k používání informačního systému

- 1) Optimalizace nákladů na běžný chod firmy.
- 2) Podpora možnosti dalšího rozšiřování, rozvoje firmy.
- 3) Podpora vytvoření (udržení) dobrého jména firmy.
- 4) Optimalizace času potřebného pro běh operací spojených s chodem firmy.
- 5) Zvýšení efektivity u prováděných činnostech. (9)

Motivace firmy k používání IS	Průměrně hodnocení faktorů
Nízká	<0,4
Střední	0,4-0,6

Vyšší	0,6-0,8
Vysoká	>0,8

Tabulka 6: Motivace firmy a hodnocení faktorů (Zdroj: (9) Vlastní zpracování)

Faktor	1	2	3	4	5	Průměr
Hodnota	0,5	0,4	0,3	0,8	0,7	0,54

Tabulka 7: Motivace firmy k používání IS (Zdroj: Vlastní zpracování)

Motivace firmy k používání informačního systému vyšla **dle tabulky č.7**, jako střední.

Náročnost na informační úroveň firmy

Díky znalosti hodnot motivace k používání IS a náročnosti na IS je možné stanovit náročnost na informační úroveň firmy.

Motivace k používání IS			Náročnost na informační systém			
			nízká	střední	vyšší	vysoká
nízká			1	2	3	3
střední			2	3	4	4
vyšší			3	4	4	5
vysoká			3	4	5	5

Tabulka 8: Náročnost na informační úroveň firmy (Zdroj: Vlastní zpracování)

Jak je patrné z tabulky, hodnota náročnosti na informační úroveň firmy je druhá nejvyšší, tedy odpovídá číslu 4.

Stanovení fáze životního cyklu IS

Existující informační systém se může nacházet ve 4 fázích životního cyklu:

- zavádění IS
- fáze růstu
- fáze zralosti
- fáze doběhu

Jednotlivé fáze mají svá specifika a na základě nich bych analyzovaný systém zařadila **na pomezí fáze růstu a fáze zralosti**, neboť jsou zde stále místa v systému, která by se dala vylepšit a do budoucna by se mohly vyskytnout procesy, které by bylo vhodné do

systemu integrovat, nicméně veškeré hlavní funkce uživatelům systém již nabízí s co největší efektivitou.

Stanovení souhrnné optimální úrovně IS

Na základě hodnoty náročnosti na informační úroveň a stanovení fáze životního cyklu IS je možné určit souhrnnou optimální úroveň IS.

Náročnost na informační úroveň firmy	Životní cyklus informačního systému				
	1	1	1-2	1-2	1
	2	1	1-2	2-3	1-2
	3	2	2-3	3-4	2-3
	4	2-3	3-4	4-5	3-4
	5	3-4	4-5	5	4

Tabulka 9: Stanovení optimální úrovně IS (Zdroj: Vlastní zpracování)

Z uvedené tabulky vyplývá, že hodnota souhrnné optimální úrovně informačního systému se pohybuje mezi 3–5, přičemž asi nejvíce inklinuje k hodnotě **4**. Z daných hodnot vyvozují **minimální** optimální hodnotu na **3,5** a **maximální** optimální hodnotu na 4,5.

3.17 Popis jednotlivých zkoumaných oblastí systému

ORGWARE

Ve firmě existují doporučené pracovní postupy, které však uživatelům nejsou dostatečně známy a uživatelská příručka pro práci se systémem není zpřístupněna širokému okruhu uživatelů.

PEOPLEWARE

Při nástupu nových zaměstnanců je každý uživatel seznámen s postupy práce s informačním systémem. Hloubka školení se odvíjí především od pozice, kterou ve firmě pracovníci zastávají. Každý zaměstnanec má určitá práva k čemu v systému smí nebo nesmí přistupovat. Uživatel je tedy zaškolen pouze na části systému, kterou během své práce potřebuje. Není tedy třeba, aby každý uživatel uměl pracovat s celým systémem. Ve společnosti nejsou pevně zavedena konkrétní školení, forma školení spočívá především v podání potřebných informací, při nástupu do zaměstnání.

DATAWARE

Systém poskytuje dostatečný objem dat, která potřebují uživatelé k efektivnímu rozhodování. Bezpečnost systému je zajištěna omezenými přístupovými právy, zálohami a existencí plánu pro obnovu dat.

SECURITY

Ve společnosti jsou prováděny pravidelné zálohy. Zálohování dat mají na starosti pověřené osoby. Provádí se zálohy všech virtuálních serverů tak i všech databází. Zálohy na externích discích jsou uloženy mimo budovu společnosti v případě možnosti požáru či jiných živelných pohrom nebo krádeží. Společnost vyvíjí maximální úsilí k eliminaci jakýchkoliv bezpečnostních hrozeb.

SUPPLIERS

Komunikace s dodavatelskými systémy, není v systému řešena. Tato oblast není primárním zájmem společnosti. Společnost vyvíjí vlastní softwarová řešení, která dále prodává svým zákazníkům. K vývoji těchto produktů využívá především vlastních lidských i materiálových zdrojů.

CUSTOMERS

Zákazníci systému informačního systému jsou samotní zaměstnanci společnosti. Samotný systém Time Misson, je však i prodejním produktem společnosti, který je v rámci poskytovaných produktů nabízen zákazníkům.

MANAGEMENT IS

Management IS sleduje dodržování pravidel práce se systémem, sleduje a podporuje další možný vývoj systému, neboť si vědom, že informační systém je důležitým zdrojem informací, která napomáhají k dalšímu možnému chodu a rozvoji společnosti.

MANAGEMENT

I tato oblast managementu podporuje rozvoj systému a hojně využívá data z něj získaná, potřebná pro důležitá rozhodnutí. Úzce spolupracuje se správcí systému.

HW a SW

Tyto oblasti jsou blíže popsány v předchozích kapitolách.

Určení váhy jednotlivých kritérií u hodnocených oblastí informačního systému a stanovení významu jednotlivých oblastí IS pro firmu

V každé zkoumané oblasti jsou 2-3 kritéria, ke kterým se vztahují jednotlivé otázky, na něž respondenti odpovídali. Nastavení vah je možné v rozmezí 1-10, přičemž platí, že 1

značí **nejnižší** možnou důležitost a **10** pak důležitost **nejvyšší**. Pro detailnější postup, jak správně dané váhy stanovit vzhledem k životnímu cyklu IS, jsem opět postupovala dle pravidel a doporučení metody HOS09.

Oblast	Otázky	Hodnota kritéria
OW	1,2,3	7
	9	7
	4,5,6,7,8,10	10
DW	1,2	9
	7,8,9	10
	3,5,6,10	10
PW	1,2,10,3	7
	4,5,6	8
	7,8,9	3
SEC	2,5,6,9	7
	3,4,7,8,10,1	10
CUS	1,2,3,4	5
	5,8	8
	6,7,9,10	10
MA	1,6,7,8,10	10
	2,5,9	8
	3,4	6
MIS	1,6,9,10	8
	2,7,8	10
	3,4,5	10
HW	1,4,6	9
	2,5,10	10
	3,8,9,7	8
SW	1,4,5,7,8	8
	9,10	7
	2,3,6	6
SU	1,2,7	5
	3,4,5,6	9
	9,9,10	10

Tabulka 10: Stanovení vah kritérií (Zdroj: Vlastní zpracování)

Posledním krokem v aplikační fázi je pak zodpovězení jednotlivých otázek vhodně vybranými respondenty ze všech zkoumaných oblastí.

3.17.1 Fáze tvorby výstupů

Výpočet hodnot stavu za jednotlivé oblasti

Výpočet hodnot zjistím, převedením textových odpovědí jednotlivých otázek do číselné podoby. Bodové hodnoty byly již stanoveny metodou HOS09. Následně jsem dosadila do **vzorce č. 1** (str. 28 uvedeného v teoretické části práce) a zjistila, tak hodnotu jednotlivých oblastí.

Zkoumaná oblast	Hodnota oblasti
OW	3,58
PW	3,46
DW	3,91
SEC	4,08
MA	4,40
MIS	3,47
CUS	3,50
HW	3,76
SW	3,31
SU	3,89

Tabulka 11:Hodnoty jednotlivých oblastí (Zdroj: Vlastní zpracování)

Zjištění hodnoty celkového stavu informačního systému

Celkovou hodnotu stavu informačního systému zjistíme, tak že vezmeme zjištěné hodnoty za jednotlivé zkoumané oblasti IS (mimo oblast HW a SW, která se analyzuje odděleně) a zjistíme jejich minimální hodnotu. Tato hodnota pak bude rovna celkovému stavu informačního systému.

Pro vyhodnocení celkové úrovně systému je klíčový předpoklad: „*Informační systém je natolik silný, nakolik silná je jeho nejslabší část.*“ (9, s. 96)

Nejslabší článek systému je definován **vzorcem č. 4.**

$$H = \min \{O_{ow}; O_{pw}; O_{dw}; O_{se}; O_{cu} O_{su} O_{mis} O_{ma}\}$$

Vzorec 4: Celkový stav informačního systému (Zdroj: (9) Vlastní zpracování)

H-celková souhrnná hodnota zkoumaného IS

O_i - jednotlivé oblasti systému

S nepatrným rozdílem mezi oblasti MIS a PW má nejnižší hodnotu oblast PEOPLEWARE 3,46.

Celková hodnota systému je tedy **H=3,46.**

3.17.2 Fáze zpracování výstupů

Fáze zpracování výstupů zahrnuje analýzu vazby na hardware a software, zjištění míry vyváženosti a optimality informačního systému a grafické zpracování analýz výsledků metody.

Analýza vzájemné vazby mezi HW a SW

Aby byl software efektivně využíván, je potřeba, aby hardware splňoval alespoň minimální požadavky na jeho funkčnost.

Výrazně efektivnější je využívat hardware vysoké úrovně tak, že je na něm využíván takový software, jehož funkčnost si vystačí i s procenty výkonu takového hardwaru. (9)

Po dosazení do **vzorce č. 2** (str. 29) vyšla hodnota vazby **-11,9**, což značí podle metody HOS09 **vyvážený stav**. Obecně to tedy znamená, že hardware není v tomto případě plně využíván používaným softwarem. Kladná hodnota by značila, že využívány hardware je nedostačující, pro firmou využívány software.

Zjištění míry optimality vyváženosti informačního systému

Pomocí následujícího vzorce stanovím optimální hodnoty jednotlivých úrovní, na základě kterých zjistím, zda jsou naměřené hodnoty pozitivní či negativní.

$$o_{vi} = o_{vmin} + V_i(o_{vmax} - o_{vmin})$$

Vzorec 5: Optimální hodnoty jednotlivých úrovní (Zdroj (9): Vlastní zpracování)

O_{vi} ... optimální hodnota i-té oblasti systému

O_{vMIN} ... optimální minimální hodnota systému

O_{vMAX} ... optimální maximální hodnota systému

V_i ... význam i-té oblasti IS pro firmu

Oblast	Optimální hodnota oblasti
OW	4,0
PW	4,0
DW	4,0
SEC	4,2
MA	4,0
MIS	4,0
CUS	4,0
SU	3,6

Tabulka 12: Optimální hodnoty oblastí (Zdroj: Vlastní zpracování)

Zjištění míry nevyváženosti informačního systému

K určení míry nevyváženosti použijí **vzorec č.3** str. 29 uvedený v teoretické části práce.

Oblast	Ni-hodnota nevyváženosti
OW	-11,30
PW	-14,02
DW	-2,83
SEC	-3,13
MA	9,34
MIS	-13,77
CUS	-13,02
SU	6,63

Tabulka 13: Nevyváženost oblastí (Zdroj: Vlastní zpracování)

Celkovou nevyváženost zjistíme, jako maximální hodnotu ze všech hodnot nevyváženosti jednotlivých oblastí v absolutní hodnotě. V případě společnosti ARTIN, je maximální hodnota nevyváženosti v **absolutní hodnotě** oblast **PW** a to **14,02%**.

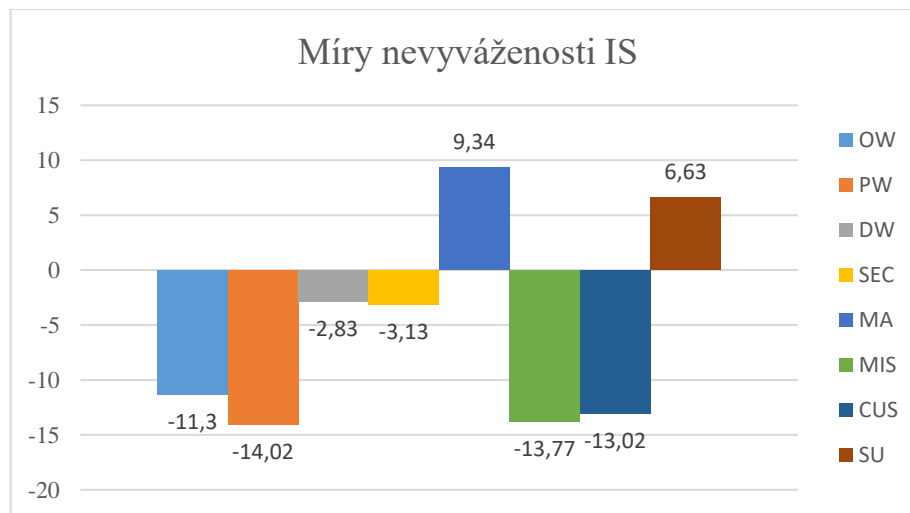
3.18 Vyhodnocení metody HOS09

V **tabulce č. 13** jsou uvedeny číselné hodnoty míry nevyváženosti informačního systému. Zkoumané oblasti se vyskytují na pomezí více různých stavů, některé oblasti jsou vyvážené některé mírně nevyvážené. Nejnížší hodnoty se vyskytují u oblastí MIS, PW a CUS. Toto zjištění nutně neznamená, že se jedná o problémový IS. Výsledky nám mohou odhalit do jakých oblastí je investováno přebytek prostředků, nebo jaké oblasti jsou na úkor jiných zanedbávány.

Z **grafu č. 1** lze vypočítat hodnoty nevyváženosti jednotlivých oblastí. Kladné hodnoty znamenají, že daná oblast je ve firmě **nadhodnocována**. Záporné hodnoty naopak znamenají, že daná oblast je ve firmě **podhodnocena**. Jako mírně podhodnocené oblasti se jeví oblast PW, MIS a CUS. Tyto výsledky se odvíjely z dotazníkového šetření, které mělo v každé oblasti zjistit určitá kritéria, předem stanovená metodou HOS09. Z výsledků oblasti CUS bylo zjištěno, že se uživatelé příliš často nepotkávají s pravidelným vyhodnocováním potřeb na informační systém. Time Mission uživatelům však umožňuje podat zpětnou vazbu vývojářům aplikace. Každý modul systému obsahuje odkaz vedoucí na issue tracking systém, kde mohou uživatelé nahlašovat defekty systému nebo naopak doporučení na zlepšení uživatelského rozhraní. Zpětná vazba je v současnosti uživateli spíše využívána na hlášení defektů systému, nikoliv na vyhodnocování potřeb uživatelů. V oblasti MA a MIS můžeme sledovat naopak odlišné výsledky, kdy oblast MIS vyšla jako částečně podhodnocena, naopak oblast MA vyšla jako mírně nadhodnocena.

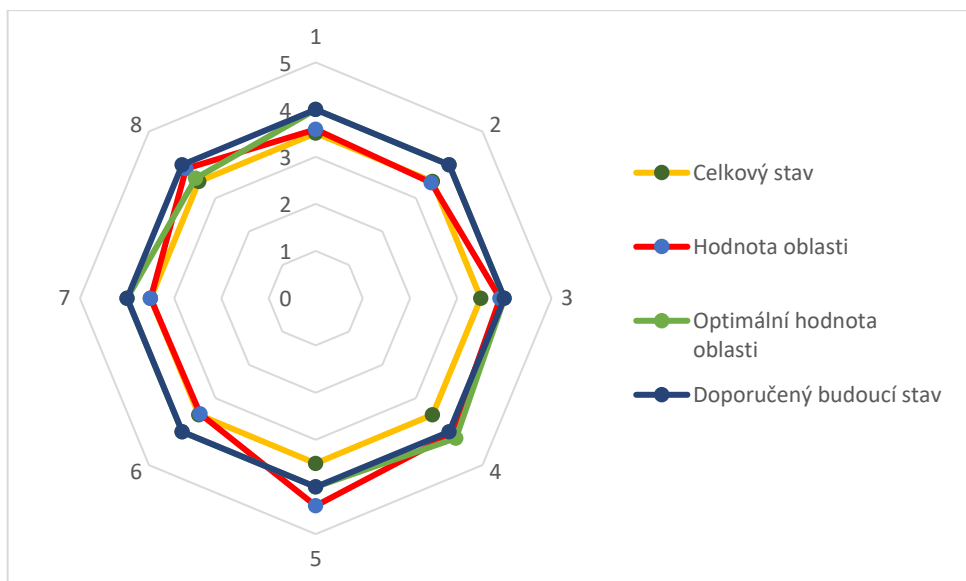
Nejblíže k vyváženosti inklinovaly oblasti security a dataware které se blížily k téměř ideálním hodnotám vyváženosti. Zaměříme-li se na technologii, pak soulad mezi **hardware** a **software** vyšel jako vyvážený, znamená to tedy, že hardware není v tomto případě plně využíván používaným softwarem. Nicméně z osobních zkušeností nevnímám, že by firmou využívaný hardware byl výrazně výkonnější, než by byla potřeba pro současně využívaný software napříč všemi různými projekty.

Celkově je dle metody HOS09 analyzovaný systém ve více oblastech mírně podhodnocen a má své nedostatky, které je potřeba řešit, aby do budoucna nedocházelo k prohlubování neefektivností, které by mohly brzdit nebo narušovat chod systému.

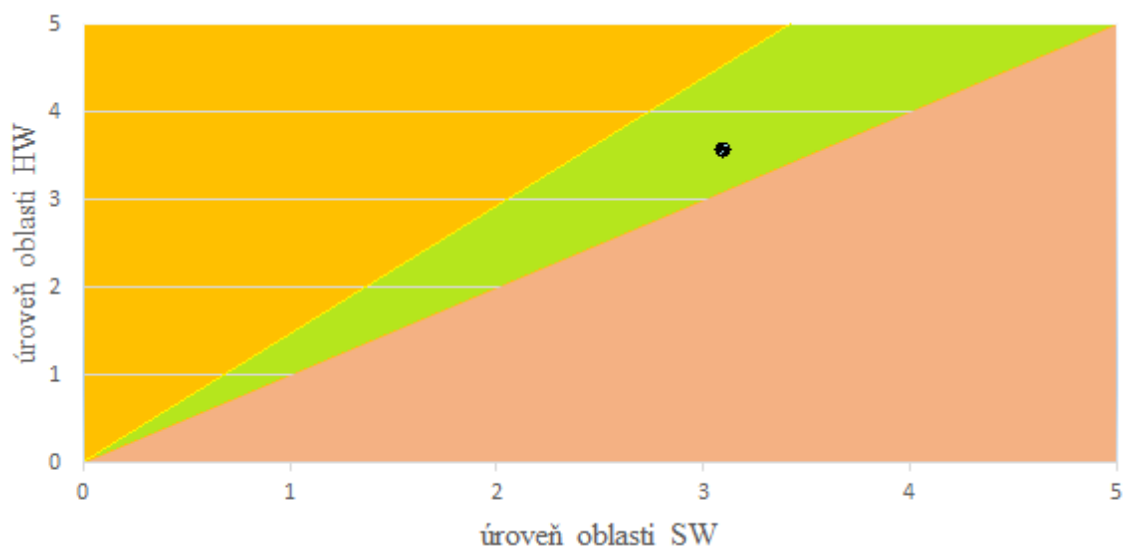


Graf 1: Míry nevyváženosti IS (Zdroj: Vlastní)

Následující paprskový graf reprezentuje celkový stav systému vůči jednotlivým skutečně naměřeným hodnotám, optimálním hodnotám a také doporučenému budoucímu stavu jednotlivých oblastí. Doporučené hodnoty jsem získala z faktu, že analyzovaný informační systém se nachází na pomezí fáze růstu a zralosti, přičemž spíše inklinuje k fázi zralosti. Optimální hodnoty byly zjištěny na základě znalosti minimální a maximální hodnoty oblasti systému a taky koeficientům významu jednotlivých oblastí pro společnost.



Graf 2: Paprskový graf zobrazující celkový stav IS (Zdroj: Vlastní tvorba)



Graf 3: Technologie informačního systému (Zdroj: Vlastní zpracování)

Porovnáním hardware a software zjistíme, že se nacházíme v zelené zóně, což tedy značí, že hardware a software jsou ve vzájemném blízkém souladu.

4 Vlastní návrhy řešení

Tato kapitola má za cíl stanovit návrhy možných řešení, které povedou ke zlepšení stávající situace IS. V kapitole se zaměřím na zavedení změn především v oblasti Software Quality assurance a na zjištěné nedostatky, které byly odhaleny na základě provedených analýz.

4.1 Doporučené strategie na základě provedené metody HOS09

Metoda HOS09, kterou jsem použila k analýze současného stavu IS, doporučuje 3 různé strategie, na základě celkového zhodnocení informačního systému. Jedná se o strategie **rozvoje**, **udržení** a **útlumu**.

Jednotlivé strategie jsou voleny v závislosti na stávajícím stavu oblasti a budoucích doporučených hodnot stavu jednotlivých oblastí. Pokud jsou tedy hodnoty jednotlivých oblastí nižší, než budoucí doporučené hodnoty aplikuje se **strategie rozvoje**, pokud jsou si hodnoty současného a budoucího stavu rovny, použije se **strategie udržení** a v případě, že jsou hodnoty současného stavu jednotlivých oblastí vyšší než budoucí doporučený stav, použije se strategie **útlumu**.

Z provedené analýzy vyplynulo, že některé oblasti činí celkový stav IS jako mírně nevyvážený (dle hodnot **tabulky č.1** uvedené v teoretické části práce).

U mírně nadhodnocených oblastí bych volila spíše strategii **udržení**. Žádná z analyzovaných oblastí nevyšla jako vysoce nadhodnocena, do žádného segmentu nejsou tedy vydávány nadbytečné výdaje, které by mohly ohrozit chod celé společnosti.

A konečně u oblastí, které byly označeny jako mírně podhodnoceny, bych volila strategii **rozvoje**. V následující tabulce bude ke každé oblasti zvolena doporučená strategie a důvody pro její zvolení. Uvedené důvody, vyplynuly z dotazníkového šetření.

Jednotlivé oblasti	Strategie	Důvod
Orgware	Rozvoj	Směrnice a postupy v případech zotavení IS z nestandardních situací nejsou pro uživatele dostatečně známe. Neprobíhá včasná dokumentace provedených změn v systému a uvedení důvodů, které ke změně vedly ani pravidelné kontroly dodržování vnitřních pracovních postupů a směrnic.
Peopleware	Rozvoj	Uživatelům IS není poskytnuta dokumentace běžných pracovních postupů práce s informačním systémem. Informační centra se nyní zabývají spíše řešením významných problémů s IS, dlouhodobé zlepšení chodu systému je spíše upozaděno.
Dataware	Udržení	V systému jsou k dispozici dostatečně kvalitní data. Případné defekty jsou včas řešeny.
Security	Udržení	Data jsou v systému dostatečně zabezpečena díky monitorování důležitých činností a prováděním pravidelných záloh pověřeným pracovníkem.
Suppliers	Udržení	V současné době není IS primárně určen ke komunikaci s dodavateli. Neboť soudobá situace firmy si to zatím nevyžaduje.
Customer	Rozvoj	Ve společnosti neprobíhá pravidelné vyhodnocování potřeby uživatelů a přínosů, které uživatelé od IS očekávají.
Management IS	Udržení	Management IS funguje dle výsledků metody správně.
Management	Udržení	Tato oblast byla vyhodnocena jako mírně nadhodnocena, přesto však nedoporučují strategii útlumu, ale spíše udržení.
HW a SW	Udržení	Vztah mezi HW a SW vyšel jako vyvážený.

Tabulka 14: Zvolené strategie jednotlivých oblastí (Zdroj: Vlastní zpracování)

4.2 Návrhy změn na základě SWOT analýzy a vlastního pozorování

Návrhy v této části budou vycházet, jak ze swot analýzy informačního systému, tak i z posouzení vlastním pozorováním.

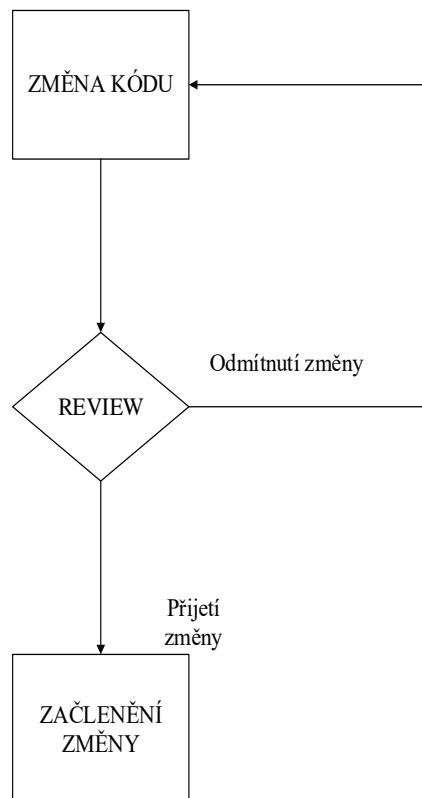
Na projektu Time Mission v současnosti pracují 3 vývojáři, z nichž jeden má hlavní odpovědnost za chod celého projektu. Je schvalovatelem různých změn a rovněž se stará i o nasazování nových verzí na produkci. Během celé doby vývoje aplikace Time Mission se na projektu vystřídal řada pracovníků. Pro většinu se tento projekt ve společnosti stal, jakýmsi pilotním projektem, na kterém rozvíjí své zkušenosti. Je tedy potřebné, aby při nástupu nových či již stávajících pracovníků docházelo patřičným způsobem k předávání znalostí v co nejkratším možném časovém intervalu. Proto se tedy v této části kapitoly budu snažit navrhnout několik možných postupů, jak toto shromažďování znalostí rozšířit.

4.2.1 Code review

Code review je proces, který zajistí, že každou změnu v kódu uvidí další osoba a může ji připomínkovat. Recenzent se tak především soustředí na správnou a logickou strukturu kódu. Aby vše mohlo dobře fungovat, je potřeba formálně zaznamenat styl a pravidla psaní kódu, na kterých se vývojáři společně dohodnou. Code reviews slouží především ke zlepšení kvality kódu, zejména jeho čitelnosti. Vývojáři budou více motivováni psát lepší kód, když ví, že někdo další bude tento kód číst a hodnotit. Připomínky mohou pomoci psát kód jednodušeji, případně mohou pomoci využívat již existujících funkcí a zamezit tak duplicitě kódu. (15)

Tento proces může být i velkým přínosem pro nově příchozí členy týmu na projektu, kdy se málo zkušený vývojář může dozvědět mnoho tipů na vylepšení psaní kódu a rychleji se tak adaptovat v nové práci.

Postup při code review. Proces má dva účastníky autora změny a recenzenta. Autor tedy vytvoří nějakou změnu, a když je změna připravena, tak recenzent provede review. Výsledkem je pak odsouhlasení změny nebo její úprava.



Obrázek 20: Postup při Code review (Zdroj: Vlastní zpracování)

Hlavní výhodou pro společnost bude ve sdílení znalostí a zkušeností mezi recenzentem a autorem. I v případě, že na místě recenzenta bude méně zkušený vývojář, dojde k přenesení znalostí.

U autora je důležité, aby poznal, kdy je změna připravena na recenzi. Změna by měla být izolovaná a měla by fungovat bez nutnosti dalších změn v kódu v daném repozitáři. Změna by měla být **malá**, například implementace požadované funkcionality nebo oprava problému. Velké změny se hůře kontrolují a jsou více nepřehledné a efektivita code review, při velkých změnách klesá. Samotná kontrola pak probíhá tak, že se změna nahraje na dohodnuté místo například na společnosti využívaný GitLab a recenzent je osloven ke kontrole. Další důležitou otázkou je, kdy dané změny začlenit, jestli je začleňovat před nebo po recenzi a kdo začlení kód po kontrole. Osobně bych se přikláněla začleňovat změnu až po recenzi. V případě nasazení kódu by tato role byla přidělena hlavnímu vývojáři Time Mission, který je pověřený změny nasazovat.

Z pohledu recenzenta je důležité, aby kódu rozuměl. Dalším důležitým bodem jsou **testy** pokud má daná změna vytvořené testy, recenzent je může využít, pokud testy nejsou, je dobré si rozmyslet, jak náročné je danou změnu testovat, zda to má smysl. V těchto případech se může vyskytovat problém v designu kódu. Pokud testy vytvořené jsou, je důležité zjistit, zda testují, co mají a jsou odolné proti budoucím změnám. V neposlední řadě si recenzent musí ověřit i skutečnost, zda změna neovlivnila další části kódu a jestli změna ostatní části kódu nerozbije.

Aby mohl proces správně fungovat, je důležité, aby komunikace mezi jednotlivými vývojáři probíhala konstruktivně a nekonfliktně. Pokud recenzent nějakou část kódu kritizuje, je dobré udat důvod proč, a jak jde daná věc napsat lépe. Není tedy dobré pouze kritizovat, ale také navrhnout možné způsoby řešení.

Existují přímo nástroje, které se specializují na code review a které nejsou závislé na firmou využívaném systému správy verzí. V praxi je často využíván nástroj **Review Board**. (15)

4.2.2 Automatické testování a integrace testů do JENKINS

Dalším možným řešením, jak zkvalitnit vývoj softwaru je jeho testování ať už manuálním nebo automatizovaným způsobem. Předchozí kapitola se zaměřovala především na kvalitu a kontrolu kódu samotnými vývojáři. Ukazuje se však, že člověk, který kód vyvinul a zná přesně funkcionality celé aplikace, má větší problém aplikaci důkladně otestovat, protože není možné se jednoduše od těchto znalostí vymanit. Autor má jasnou představu, jak by se daná funkcionality měla chovat a čemu se tak vyhnout. Uživatelé systému však nemusí určité postupy dodržet a mohou tak systém dostat do stavu, na který není připraven. Ve společnosti je v současné době na projektu zavedeno manuální testování, které provádí pověření pracovníci. Tito zaměstnanci testují převážně opravy stávající funkcionality. V případě zavedení nových funkcionalit se přechází i na testování nových částí systému.

Jedním z možných řešení, jak tuto práci více zefektivnit a zrychlit je právě automatizované testování. K tomuto účelu lze využít několik různých automatizačních nástrojů. Samotná automatizace může ušetřit čas během častých úkonů, které souvisí s manuálním testováním, jako například generování testovacích dat, extrakce a transformace výstupních dat, či porovnávání výsledků pomocí maker v excelu nebo jiných nástrojů.

Motivace pro automatické testování

- Odstranění opakujících se úkolů
- Úspora času, tester se tak může věnovat činnostem, které si žádají větší soustředění a přináší větší užitek
- Pomáhá eliminovat chyby z nedbalosti

Příprava automatických testů

- 1) Výběr vhodného nástroje
- 2) Návrh testů
- 3) Stanovení priorit: Určíme si kritické testy, jejichž splnění má pro nás nejvyšší prioritu, stejně tak u ostatních testů je důležité stanovit priority. Následně je nutné provést kontrolu, že námi nastavené priority se shodují s představami vývojářů.
- 4) Spuštění automatických testů, které mohou být využity jako šablona pro další podobné testy.
- 5) Spuštění testů dle priorit.
- 6) Ověřit, jak velké je pokrytí kódu automatickými testy.
- 7) Dokumentace provedených testů.

Automatizační nástroje

Ve společnosti jsou v současné době automatizovány pouze jednotkové testy, kterými se budu blíže věnovat v následující části. K zefektivnění současného manuálního testování, by tedy bylo vhodné automatizovat i regresní testování pro grafické uživatelské rozhraní. K tvorbě těchto testů lze využít řadu nástrojů, z nichž některé jsou i volně dostupné.

Správný výběr nástroje závisí na úspěšnosti celé automatizace. Prvním důležitým kritériem je plná podpora technologie. Tím lze do určité míry zmírnit riziko, že během testů bude odhaleno, že určitý element podporován není a vytvořený test, tak nebude funkční.

Dalším důležitým faktorem je i cena, některé nástroje jsou však volně dostupné a v našem případě i vyhovující, neboť některé nástroje pro automatizaci byly ve společnosti již využity při práci na jiných projektech.

Selenium

Automatizační nástroj pro automatické testování webových aplikací. Skládá se z více vzájemně doplňujících komponent Selenium IDE, Selenium RC, Web driver a Grid.

Je napsán v programovacím jazyce Java a lze ho využít na několika platformách. (21)

Nejčastěji je využíván Selenium IDE. Je volně dostupný, k jeho využití stačí pouze instalovat plugin do prohlížeče Mozilla Firefox. Tento nástroj využívá techniku play and record. Tato technika je založena na zachycení aktivity uživatele provádějícího v rozhraní aplikace jednotlivé testovací případy a následně umožňuje jejich přehrávání. Pokud by se jednalo o složitější testy, doporučila bych tento nástroj využívat pouze na prototypování testů a následně bych testy rovnou exportovala do Javy a zde je udržovala.

Tesabot

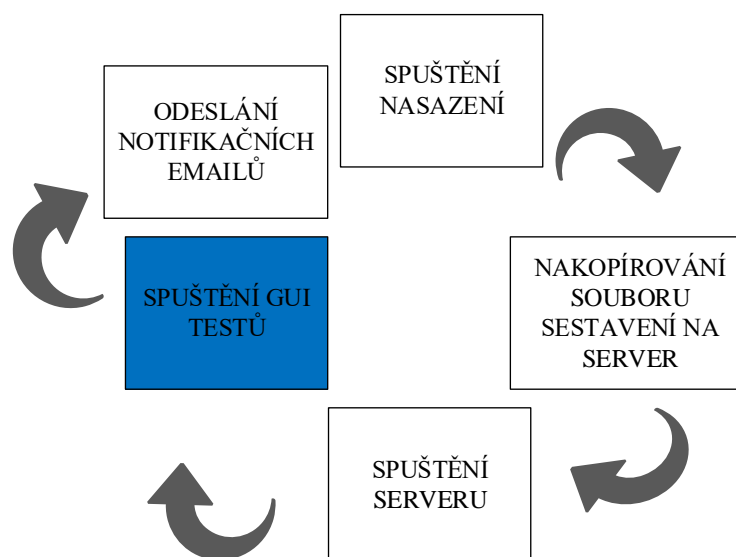
Tesabot je testovací framework, stejně jak selenium IDE využívá techniku record and play. Je stále ve fázi vývoje, na kterém pracuje Brněnska společnost Tesabot. Vyvíjen je opět k automatizaci webových aplikací, využívá strukturu data object model k orientaci mezi prvky. K připojení na stránku využívá proxy server.

JMeter

Nástroj primárně určen pro performance (zátěžové) testy a load testy webových aplikací. JMeter umožňuje vytvářet specifické testy pro aplikace, které simulují reálnou zátěž během provozu systému. Lze nastavit libovolný počet uživatelů a odesílat jejich požadavky na konkrétní objekty.

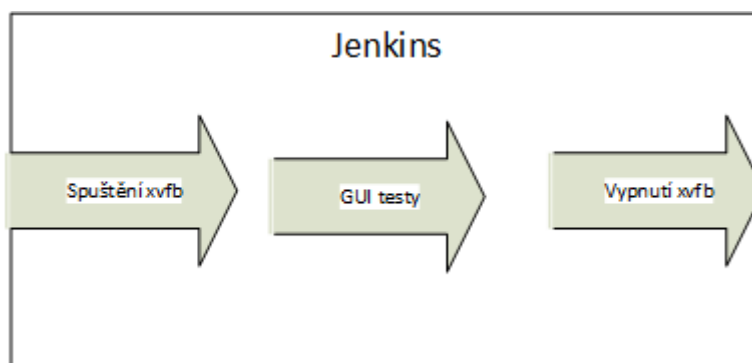
Integrace testů do JENKINS

Jak jsem již zmiňovala ve slabých stránkách prostřednictvím SWOT analýzy, ve firmě již existují automatické testy pro GUI, pokrývající velkou část systému. Tyto testy však nebyly využívány a ani dlouhodobě udržovány, řada z nich je již nyní nefunkčních, například z důvodu změn ve funkcionalitě aplikace. Doporučila bych tedy testy opravit a integrovat na build server, kde by se pravidelně po nasazení nové verze či jakékoliv jiné úpravě ve funkcionalitě kódu automaticky spouštěly po každém nasazeném buildu na testovací prostředí. Proces **nasazení sestavení** by se pak následně rozšířil o další činnost, spuštění GUI testů viz následující obrázek.



Obrázek 21: Proces nasazení sestavení po implementaci GUI testů (Zdroj: Vlastní zpracování)

Po úspěšné spuštění testů na serveru, je ještě potřeba instalovat do Jenkins plugin, který bude spouštět a vypínat Virtuální framebuffer. Tento virtuální framebuffer nahrazuje výstupní zařízení, jenž na serveru není dostupné.



Obrázek 22: Proces spuštění testů GUI testů pomoci XVFB (Zdroj: Vlastní zpracování)

4.2.3 Dokumentace testů za pomoci snímkování

Po úspěšné integraci testů do Jenkins by bylo užitečné do již vytvořených testů vložit do kódu funkci, která by během neúspěšného testu zachytila snímkem přesné místo, kde daný test selhal a ukládala tyto snímky na předem zvolené místo.

4.2.4 Unit testy

Jak jsem již popisovala v teoretické části této práce, unit testy slouží převážně k testování jednotek zdrojového kódu. Obecně by psaní unit testů mělo fungovat tak, že programátor napíše metodu a bezprostředně nato pro tuto metodu napíše test. Psaním těchto testů získá společnost přehled během zpětné kontroly produktu, po jakékoliv změně v aplikaci. Testy jsou většinou naplánovány na periodický běh například každou noc. Testovací sadu lze také nakonfigurovat na automatické spouštění při každé změně hlavního kódového uložště.

Současná situace unit testů na projektu Time Mission

Současná testovací sada unit testů pokrývá **odhadem** 40% celé funkcionality. Veškeré testy jsou psány, tak že prvně je napsána přímo funkcionalita a následně na to jsou napsány testy. Testy jsou vytvořeny především na složitější metody. Nelze nyní říci, zda je toto pokrytí dostatečné.

Jednotkové testy by měli splňovat několik důležitých podmínek:

- Měly by být opakovaně vykonatelné
- Jednoduché na implementaci
- Testy by se neměly v budoucnu měnit
- Každý by měl být schopen testy spouštět
- Běh testů by neměl být dlouhý
- Testy by měly pokrývat většinu funkcionality

Velikost pokrytí funkcionality testy není přesně stanoveno, u každé aplikace může být tato hodnota rozdílná. Doporučila bych tedy implementaci pluginů do build serveru Jenkins, které dokážou přesně říci, jak velká část kódu je testy pokryta a doporučit vhodnou velikost pokrytí testů konkrétní aplikace. Jedná se o pluginy Configuring the test coverage metrics in Jenkins a Adding Code Coverage and Other Metrics

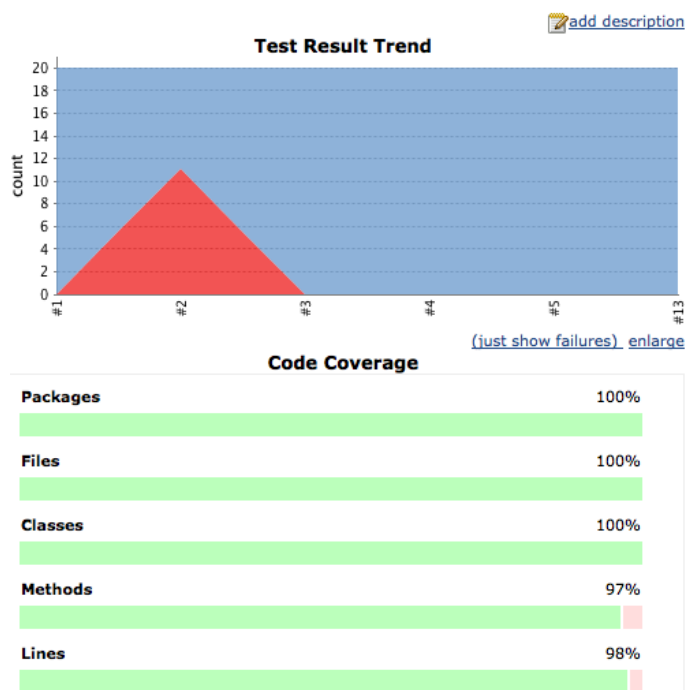
Adding Code Coverage and Other Metrics

Pomocí tohoto pluginu lze **přesně** zjistit, jak velká část kódu aplikace je pokryta testy, a které konkrétní části kódu, nejsou pokryty testovacími sadami. Stejně tak umožňuje podat informace o správném využívání postupů psaní jednotkových testů formou Test Driven Development nebo Behavior-Driven Development. (17)

Configuring the test coverage metrics

Tento plugin umožňuje nastavit i postupné zvyšování minimální hranice tolerovaného pokrytí testy, tak aby bylo stále zajištěno vysoké pokrytí v jednotlivých verzích aplikace. Pokud tedy dojde k větším změnám ve funkcionalitě, tak již vytvořené testy nemusí být funkční a také se může výrazně snížit celkové pokrytí. (17)

Oba tyto pluginy poskytují užitečné reporty se všemi potřebnými informacemi včetně jejich grafického znázornění. Například si lze zobrazit pokrytí jednotlivých metrik kódu.

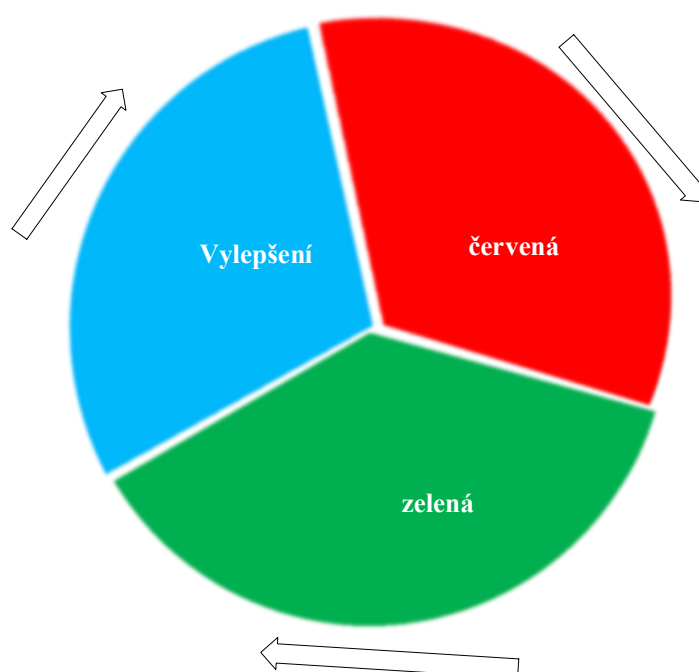


File Coverage Summary

Name	Classes	Methods	Lines	Conditionals
Cell.java	100% <div><div>1/1</div></div>	67% <div><div>2/3</div></div>	71% <div><div>5/7</div></div>	50% <div><div>3/6</div></div>

Obrázek 23: Metriky pokrytí kódu (Zdroj:(17))

Dále bych doporučila využívat i postup psaní jednotkových testů formou Test Driven Development, kdy se nejprve **definuje** funkcionalita, na ni se napíše testy, které tuto funkcionalitu ověří, poté přichází na řadu psaní kódu a nakonec úprava tohoto kódu. Tento přístup se skládá ze tří navazujících kroků. V první fázi se napíše jednoduchý test na část budoucí funkcionality, který při spuštění selže (zobrazí se červený výstup). Následně na to programátor vyvine logiku vlastní aplikace, aby dosáhl zeleného čistého běhu testu. Kód pak může následně vylepšit, při čemž kontroluje, že test stále prochází. Následně pokračuje na další část funkcionality.(10)



Obrázek 24: Test Driven Development (Zdroj: Vlastní zpracování)

Celý proces nabízí mnoho výhod. Některé jsou obecně a snadno popsatelné, jiné jsou zase subjektivní. Níže budou popsány některé z nich.

Nezávislost: Při testování existuje mnoho postupů, jak napodobit závislosti testovaného modulu. To umožňuje jej vyvinout ještě před implementací oné závislosti.

Lépe promyšlený návrh kódu: Zahájení vývoje aplikace psaním jednotkových testů vede autora kódu, k tomu, aby se nejdříve zamyslel nad výsledným konceptem funkcionality a dobře si rozvrhl jednotlivé kroky na malé dobře testovatelné části.

Možnost rychlé nápravy: Pokud se v aplikaci objeví nějaká chyba, je časově i finančně nejméně náročně tuto chybu opravit co nejdříve. Spuštění testovací sady umožňuje odhalit tyto chyby ještě ve vývoji.

Rychlá oprava i vývoj: Jednotkové testy testují především vždy malou část funkcionality, lze tedy velice snadno odhalit chyby a opravit je. Vytvořený testovací kód neslouží pouze pro testování, ale i pro ověření funkčnosti inkrementálních změn. Programátor, tak nemusí ztrácet čas ovládáním aplikace pomocí uživatelského rozhraní, aby opakovaně napodobil testovací případ, který zrovna potřebuje, ale pouze si na něj spustí již existující jednotkový test. (10)

4.2.5 Kontejnerová virtualizace

V rámci velkého množství projektů, na kterých společnost pracuje, bych doporučila využívat dockery(kontejnery) jako virtualizaci.

Kontejnerová virtualizace nebo také virtualizace na úrovni operačního systému. Jedná se o virtualizaci, při které nevzniká nový virtuální počítač, ale uzavřené prostředí, které se nazývá kontejner. Do kontejneru si lze vkládat jednotlivé aplikace (zpravidla vždy jeden kontejner jedna aplikace) a mít tak všechny knihovny, frameworky a databáze spolu s danou aplikací na jednom místě, jde v podstatě i izolaci veškerých procesů, které souvisí s daným projektem.

Hlavní výhody Dockeru

- Pomoc s integračním testováním
- Sjednocení verzí a konfigurace služeb
- Síťová virtualizace mezi kontejnery
- Usnadnění vývoje
- Rychlejší spuštění starších aplikací
- Sdílení stavů aplikací a databází mezi vývojem a testem (18)

Sjednocení verzí

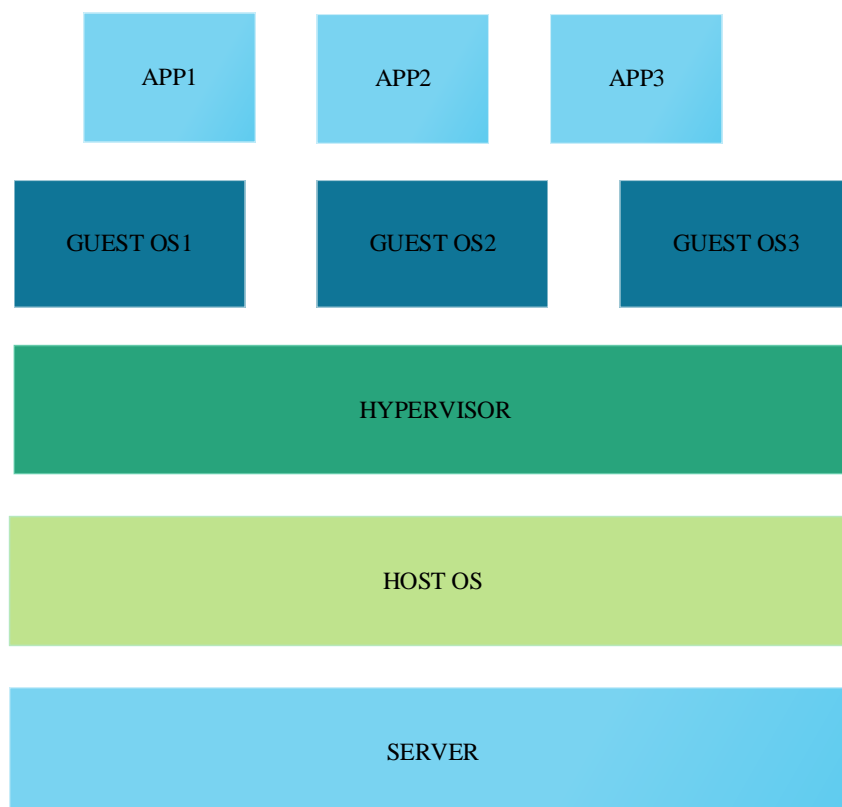
Pokud jednotlivé aplikace běží na starších a rozdílných verzích Javy nebo starších verzí MySQL, může dojít ke komplikacím při spuštění těchto aplikací na platformách jako je Mac OSX nebo jiné OS. Pokud na takových aplikacích probíhal vývoj i nyní, musel by vývojář používat verzi nástroje jinou, než na které jede produkce aplikace. Zmíněné dockery umožňují používat stejnou verzi nástroje na produkci i pro testy a vývoj.(19)

Integrační testování

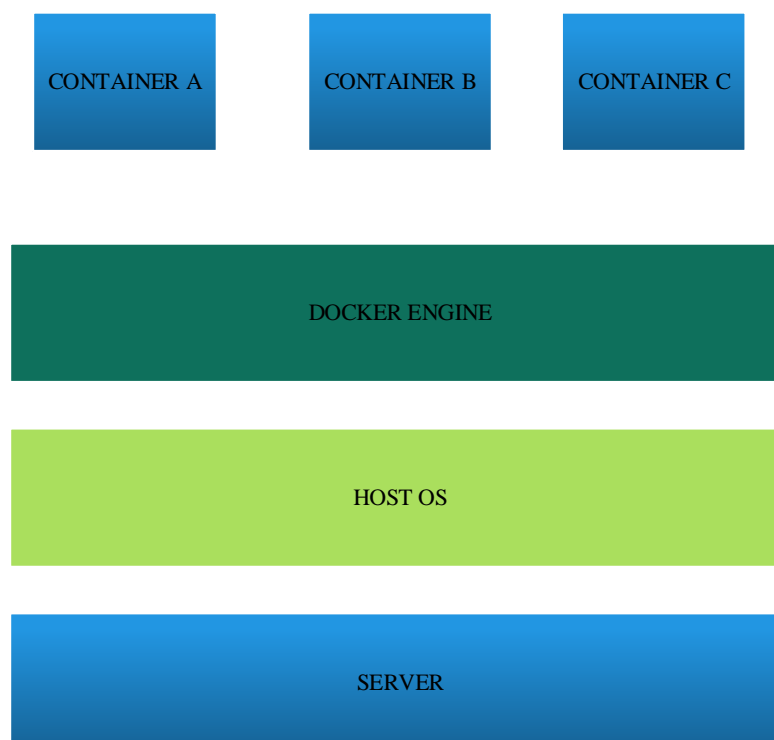
Docker umožňuje pro každý test spustit novou instanci databáze, která bude v přesně definovaném stavu, není tedy třeba řešit rozpory mezi lokální databází, kterou má uživatel nainstalovanou na svém klientském počítači či společnou sdílenou databázi. Jediný nutný krok při využití dockeru v testech je, zajistit před spuštěním testů nastartování kontejneru s databází a testy nasměrovat na tuto databázi. (19)

Úspornost

Obrovskou výhodou kontejnerů je právě jejich úspornost, na každém serveru je možné spustit 2-3 krát více než na klasických virtuálních strojích. Virtualizovaná zařízení byla dlouhou dobu řešena pomocí hypervizorů, které umožňují běh celých virtuálních strojů. Dnes se k tomuto účelu používá propracovanější forma, jako například hypervizor Xen, který běží na většině linuxových serverech. Hypervizorový přístup však spotřebovává mnoho systémových zdrojů, každý virtuální stroj obsahuje operační systém včetně celého jádra, každá jeho kopie zabírá operační paměť a uložení obdobnými soubory. Tento problém se snaží řešit právě kontejnery, které vychází z technologie FreeBSD Jails. Jails umožňuje spouštět programy v sandboxu, který nad rámec systémového jádra zpřístupňuje jen velmi omezené systémové prostředky. V následujícím obrázku bude zobrazena rozdílná architektura **virtualizace** a **dockerizace**. (19)



Obrázek 25: Virtualizace (Zdroj: Vlastní zpracování)



Obrázek 26: Dockerizace (Zdroj: Vlastní zpracování)

4.2.6 Multijazyčnost

Informační systém Time Mission již nabízí podporu pro použití více jazyků. V současné době jde přepínat na anglický jazyk, ale pouze ve staré verzi, která byla psána v JavaServerPages, kde je dnes touto technologií aktivně využívám pouze modul na zadávání času, ostatní moduly jsou dnes zobrazovány v nové verzi napsané v angularuJS. Nová verze nabízí podporu v anglickém jazyce pouze v sekci tisku u importů do XLS souborů. Přepnutí celého systému do anglického jazyka, tak není momentálně dostupné.

Vzhledem ke skutečnosti, že je systém Time Mission i prodejním produktem společnosti a je nabízen zákazníkům, kteří však nemusejí být pouze z České republiky, doporučila bych při vývoji softwaru pro nové klienty, navrhnout systém správy překladů a zvýšit tak podporu multijazyčnosti systému. Cílem by tedy bylo přesunout všechny textové řetězce zobrazované uživateli mimo zdrojový kód. V samotném kódu tak budou pouze identifikátory řetězců a v externím souboru se budou formulovat konkrétní texty pro zvolený jazyk. Použitím externího souboru se tak dosáhne zobrazení v požadovaném jazyce. Přidáním nového jazyka do systému, tak bude znamenat, pouze vytvoření nového souboru s překlady.

4.3 Odhad nákladů

Tato část obsahuje hrubý odhad nákladů jednotlivých návrhů. V **tabulce č. 15** jsou uvedeny návrhy a jejich nacenění při zpracování daných návrhů přímo vývojářem. Jednotlivé částky vychází z hrubé mzdy (44820Kč/měsíc), která byla zjištěna dle zdroje (20) jako průměrná mzda java vývojáře, s délkou praxe, vzděláním a dalšími parametry odpovídajícími současnému zaměstnanci společnosti. Jednotlivé doby odhadů byly stanoveny po konzultaci s vývojářem.

Návrh	Doba v člověkodnech	Odhad nákladů
Multijazyčnost	15 ČLD	22410Kč
Pluginy Jenkins	Maximálně 0,5 ČLD	747 Kč
Nasazení TM do dockeru	Maximálně 2 ČLD	2988Kč
Snímkování	Maximálně 1/8 ČLD	186 Kč
Údržba a refactoring testů	5 ČLD	7470 Kč
Integrace testů	2ČLD	2988 Kč

*Školení zaměstnanců s automatizačními nástroji	30 ČLD	2000Kč-75000Kč
--	--------	----------------

Tabulka 15: Odhad nákladů návrhů (Zdroj: Vlastní zpracování)

*Doba a cena školení zaměstnanců může být rozdílná, záleží zde na několika parametrech.

- Předchozí zkušenosti zaměstnanců s automatizačními nástroji
- Předchozí zkušenosti zaměstnanců s programovacími technikami
- Forma školení: Školení by mohlo probíhat ve více formách do celkové sumy lze počítat **online kurzy** nebo také **osobní školení** externí osobou či školení samotným zaměstnancem firmy, který má již s danými nástroji bohaté zkušenosti.

Cena online kurzů na automatizační nástroje se může pohybovat v přepočtu kolem 2000Kč/kurz, lze však kurzy mnohdy pořídit i za nižší akční cenu dle portálu Udemy. (22)

V případech osobního školení se může jednat o částku 300Kč/hod. a více.

Náklady spojené s automatizačními nástroji

V tabulce č. 16 jsou pak vyčísleny ceny automatizačních nástrojů.

Nástroj	Cena
Selenium	0 Kč volně dostupné
Tesabot licence 1 pracovník (měsíčně)/lze využít i zkušební verzi zdarma	0-2200Kč (zdroj)
Jmeter (open source)	0 Kč volně dostupné

Tabulka 16: Odhad nákladu automatizačních nástrojů (Zdroj: Vlastní zpracování)

Celkové náklady navržených opatření mohou dosahovat až **111789 Kč** přičemž, je tato částka především ovlivněna formou školení.

Závěr

Cílem diplomové práce byla analýza informačního systému a návrh možných změn. K jeho dosažení bylo provedeno několik dílčích analýz, a to jak samotného systému, tak i dalších technických prostředků, které souvisí s vývojem samotného systému.

Z provedených analýz vyšlo najevo několik klíčových oblastí, kde lze najít prostor vedoucí ke zlepšení. Změny v procesu vývoje by mohly vést ke zvýšení kvality systému Time Mission.

Hlavní navrženou změnou bylo zefektivnění manuálního testování za využití automatického testování a nástrojů k tomu určeným a také automatizace spouštění těchto testů v předem určeném periodickém běhu. Další změny byly zaměřeny na pracovní postupy v oblasti testování a vývoje kódu a také využití nových technologií.

Všechny navržené změny byly definovány, jako izolované změny, což umožňuje implementaci pouze některé z nich. Z hlediska ekonomického zhodnocení, jsem došla k závěru, že společnost může očekávat náklady v rozmezí 38789 Kč až 111789 Kč v závislosti na zvolené formě školení zaměstnanců.

Cíle práce tak považuji za splněné. Většina navržených zlepšení vychází z obecných praktik současných softwarových společností, která tak mají i širší uplatnění, jenž lze aplikovat i na jiné projekty ve společnosti.

Seznam použité literatury

Knihy

- 1) BASL, J. a R. BLAŽÍČEK. *Podnikové informační systémy*. 2. vyd. Praha: Grada publishing, 2007. 288 s. ISBN 978-80-247-2279-5.
- 2) KOCH, M. a J. DOVRTĚL. *Management informačních systémů*. 1.vyd. Brno: Akademické nakladatelství CERM. 2006.174 s. ISBN 80-214-3262-4.
- 3) WASSON, CH. *Systém analysis,design, and developmnet: concept principles and practices*. New Yersey: John Wiley & Sons, 2006. 832 s. ISBN 13-978-0-471-39333-7.
- 4) MOLNÁR, Z. *Moderní metody řízení informačních systémů*. 1. vyd. Praha: Grada Publishing, 1992. 347 s. ISBN 80-85623-07-2.
- 5) TVRDÍKOVÁ, M. *Zavádění a inovace informačních systémů ve firmách* vyd.Praha: Grada, 2000,109 s. ISBN 80-7169-703-6.
- 6) BLAŽKOVÁ, M. *Marketingové řízení a plánování pro malé a střední firmy*. Praha: Grada Publishing, 2007. 278 s. ISBN 978-80-247-1535-3.
- 7) SODOMKA, P. a H. KLČOVÁ. *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd. Brno: ComputerPress, 2010. 501 s. ISBN 978-80-251-2878-7.
- 8) ROUDENSKÝ, P. a A. HAVLÍČKOVÁ. *Řízení kvality softwaru*. 1.vyd. Brno: Albatros Media, 2013, 208 s. ISBN 978-80-251-3816-8.
- 9) NEUWIRTH, B. *Problematika hodnocení optimality a vyváženosti podnikových IS*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2009. 150 s. Vedoucí dizertační práce doc. Ing. Miloš Koch, CSc.
- 10) BECK, K. *Test Driven Development: By Example*. Addison-Wesley, 2003. ISBN 978-0321146533.
- 11) RAIS, K. *Risk management: studijní text pro kombinovanou formu studia*. 1.vyd. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3510-0.

Internetové zdroje

12)ARTIN.*O nás*.artin.cz[online]. ©1998-2017 [cit.2017-01-10]. Dostupné z:

www.artin.cz/o-nas

13) JUSTICE. *Výpis z obchodního rejstříku*. [online]. ©2015 [cit.2017-01-10].

Dostupné z:

<https://or.justice.cz/ias/ui/rejstrik-frima.vysledky?subjektId=205652&typ=PLATNY>

14)WEB INTEGRATION. *Verzovací systém a webintegrační projekty*. [online]. ©2012 [cit.2017-03-10]. Dostupné z:

<http://www.web-integration.info/cs/blog/verzovaci-system-a-webintegracni-projekty/>

15) ZDROJAK.*Code reviews v praxi*. [online]. © 2014-2017 [cit.2017-04-30].

Dostupné z:

<https://www.zdrojak.cz/clanky/code-reviews-praxi/>

16)ATLASSIAN.*Confluence*. [online]. © 2017 Atlassian [cit.2017-04-10]. Dostupné z:

<https://www.atlassian.com/software/confluence>

17)JENKINS. *Continuous integration with hudson*. [online]. ©2010 [cit.2017-04-21].

Dostupné z: <http://jenkins-le-guide-complet.github.io/continuous-integration-with-hudson.pdf>

18)COMPUTERWORLD.*Kontejnery ulehčí virtualizaci linuxu*. [online]. ©2016

[cit.2017-04-18]. Dostupné z:

<http://computerworld.cz/software/kontejnery-ulehci-virtualizaci-linuxu-51044>

19)ZDROJAK.*Dockerizace maven testů v CI Jenkins*. [online]. © 2014-2017

[cit.2017-04-10]. Dostupné z: <https://www.zdrojak.cz/clanky/dockerizace-maven-testu-v-ci-jenkins/>

20) PLATY.CZ. *Programátor Java*. [online]. [cit.2017-05-01]. Dostupné z:

<http://www.platy.cz/employee/viewresults/type/survey>

21) SELENIUM. *Selenium projects*. [online]. [cit.2017-04-11]. Dostupné z:

<http://www.seleniumhq.org/projects/>

22) UDEMY. *Software testing*. [online]. © 2017 [cit.2017-05-04]. Dostupné z:

<https://www.udemy.com/courses/development/software-testing/>

Osobní sdělení ve firmě

23)TOMEK, E. *Interview*. ARTIN, Božetěchova 19, 612 00 Brno. 9.3.2017.

24)KONEČNÝ, L. *Interview*. ARTIN, Božetěchova 19, 612 00 Brno. 15.2.2017.

Seznam obrázků

Obrázek 1: Vztah dat a informací	14
Obrázek 2: Čtyřvrstvá organizační pyramida z pohledu práce IS/IT v podniku	17
Obrázek 3: V-model	21
Obrázek 4: Vodopádový model	22
Obrázek 5: Analýza 7S	25
Obrázek 6: Oblasti hodnocení IS metodou HOS09	27
Obrázek 7: Organizační struktura	33
Obrázek 8: SWOT analýza informačního systému	37
Obrázek 9: Schéma zapojení počítačové sítě	39
Obrázek 10: WebServer architektura	41
Obrázek 11: Skriptovací serverové technologie JSP.....	42
Obrázek 12: Skriptovací technologie AngularJS	43
Obrázek 13: Vývojový diagram schvalovacího procesu dokumentu.	46
Obrázek 14: Jednotlivé moduly systému Time Misson	47
Obrázek 15: Životní cyklus ISSUE v JIRA software	49
Obrázek 16: Možnosti využití aplikace Confluence	50
Obrázek 17: Proces ukládání sobourů do repozitáře v GitLab.....	52
Obrázek 18: Proces buildu	54
Obrázek 19: Proces nasazení	54
Obrázek 20: Postup při Code review	71
Obrázek 21: Proces nasazení sestavení po implementaci	75
Obrázek 22: Proces spuštění testů GUI testů pomoci XVFB	75
Obrázek 23: Metriky pokrytí kódu	77
Obrázek 24: Test Driven Development	78
Obrázek 25: Virtualizace	81
Obrázek 26: Dockerizace	81

Seznam tabulek

Tabulka 1: Popis souhrnných stavů zkoumaného IS	31
Tabulka 2: Ukázka aplikací virtuálních serverů	38
Tabulka 3: Přehled počtu respondentů v jednotlivých oblastech	56
Tabulka 4: Náročnost na IS a jeho hodnocení	57
Tabulka 5: Náročnost na IS firmy	57
Tabulka 6: Motivace firmy a hodnocení faktorů	58
Tabulka 7: Motivace firmy k používání IS	58
Tabulka 8: Náročnost na informační úroveň firmy	58
Tabulka 9: Stanovení optimální úrovně IS	59
Tabulka 10: Stanovení vah kritérií	61
Tabulka 11: Hodnoty jednotlivých oblastí	62
Tabulka 12: Optimální hodnoty oblastí	64
Tabulka 13: Nevyváženost oblastí	64
Tabulka 14: Zvolené strategie jednotlivých oblastí.....	69
Tabulka 15: Odhad nákladů návrhů	83
Tabulka 16: Odhad nákladu automatizačních nástrojů	83

Seznam vzorců

Vzorec 1: Výpočet stavu jednotlivých oblastí IS	28
Vzorec 2: Výpočet míry nevyváženosti softwaru a hardwaru	29
Vzorec 3: Výpočet hodnot míry nevyváženosti jednotlivých sledovaných oblastí IS ...	29
Vzorec 4: Celkový stav informačního systému	63
Vzorec 5: Optimální hodnoty jednotlivých úrovní	64

Seznam grafů

Graf 1: Míry nevyváženosti IS	66
Graf 2: Paprskový graf zobrazující celkový stav IS	67
Graf 3: Technologie informačního systému	67

Seznam příloh

Příloha č.1 Dotazníky metody HOS2009

Příloha č.1 Dotazníky metody HOS2009

https://docs.google.com/forms/d/1aSE5f2BHf0_9z0JVnNZCSBkMfraT8DSNFjAEoiaQtsg/edit

https://docs.google.com/forms/d/1bHLRehVAc7iQE51pzMjn7UJd6KZ6YRdW9FCCweEMEio/edit?usp=drive_web

https://docs.google.com/forms/d/15ILN00u5t69RG1rk_z4gb3DCFv05ftEXUIvVsOxhnHY/edit?usp=drive_web

https://docs.google.com/forms/d/1aE_xjb2sG4XZODH6Bu1-BywqBD97iaKa93px9WhpEcg/edit?usp=drive_web